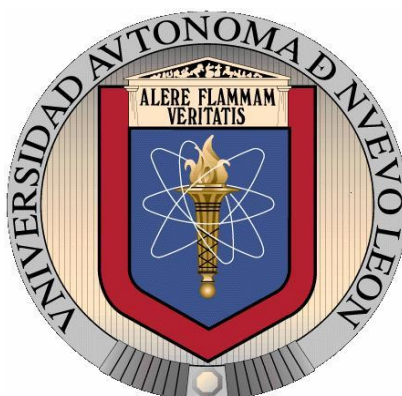


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS



ANÁLISIS PARA EL PROCESAMIENTO DE NUBES
DE PUNTOS A PARTIR DE PERFILES
BIDIMENSIONALES

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS CON ORIENTACIÓN EN MATEMÁTICAS

PRESENTA

M.E.C. GERARDO ARMANDO HERNÁNDEZ CASTORENA

San Nicolás de los Garza, Nuevo León

Enero de 2015

RESUMEN

En el presente trabajo se presenta un análisis para construir un modelo tridimensional de una pieza sólida a partir de la integración de los perfiles bidimensionales aportados por la interfase de un escáner láser, el cual hubo sido acoplado a un brazo robótico, y empleando cuaterniones para la descripción espacial del ensamble. Este ensamble escáner - robot está ideado para asistir en los procesos de inspección de las industrias manufactureras. Se presenta además un análisis, basado en el análisis de componentes principales ponderado (WPCA) combinado con el algoritmo $k - \text{means}$, para discriminar los puntos atípicos que aparecen de manera inherente en los perfiles aportados por la interfase del escáner láser, con lo cual es posible, disminuir la carga computacional del procesamiento al reducir la nube de puntos siguiendo la tendencia lineal de ciertos bloques de puntos.

Palabras clave: *cuaterniones, análisis de componentes principales ponderado, algoritmo $k - \text{means}$, nubes de puntos, puntos atípicos.*

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN

| | |
|---|---|
| 1.1 Introducción | 1 |
| | 2 |
| 1.2 Antecedentes | |
| 1.3 Motivación | 4 |
| 1.4 Aportaciones | 5 |
| 1.4.1 Algoritmo para la reconstrucción 3D | 5 |
| 1.4.2 Algoritmo para la estandarización de perfiles | 5 |
| 1.5 Organización de la Tesis | 6 |

CAPÍTULO 2. ASPECTOS TÉCNICOS DEL EQUIPO

| | |
|--|----|
| 2.1 Escáner láser | 7 |
| 2.1.1 Aspectos básicos de sistemas de escaneo láser. | 7 |
| 2.1.2 Escáner Micro – Épsilon © | 10 |
| 2.2 Descripción del sistema electromecánico. | 11 |

CAPÍTULO 3. MODELACIÓN MATEMÁTICA

| | |
|--|----|
| 3.1 Planteamiento del proceso | 14 |
| 3.2 Generalidades de los Cuaterniones | 17 |
| 3.3 Relación entre las coordenadas del escáner láser y el sistema coordenadas global. | 21 |
| 3.4 Algoritmo para la construcción de un sólido geométrico a partir de la integración de perfiles bidimensionales. | 29 |
| 3.5 Análisis y procesamiento de datos. | 32 |
| 3.5.1 Agrupamiento o clusterización por medio del algoritmo k – means. | 33 |
| 3.5.2 Análisis de componentes principales y una variante ponderada. | 35 |
| 3.5.3 Estimación de la tendencia en cada grupo. | 37 |

| | |
|--|--------|
| 3.5.4 Reducción de la nube de puntos. | 39 |
| CAPÍTULO 4. EXPERIMENTACIÓN | 40 |
| 4.1 Ejemplo numérico del uso de cuaterniones e integración de un sólido geométrico. | 41 |
| 4.2 Discriminación de puntos atípicos (outliers) y reducción de una nube de puntos. | 52 |
| 4.3 Dificultades técnicas en la reconstrucción tridimensional. | 56 |
| CAPÍTULO 5. RESULTADOS Y CONCLUSIONES | 61 |
| 5.1 Conclusiones | 61 |
| 5.2 Recomendaciones y trabajo futuro. | 62 |
| 6. ANEXOS | 63 |
| 6. 1 Anexo #1: Rutina en Matlab© para cambiar la orientación espacial usando cuaternios. | 63 |
| 6.2 Anexo #2: Rutina en Matlab © para la reconstrucción 3D a partir de perfiles bidimensionales. | 65 |
| 6.3 Anexo #3: Rutina en Matlab © para obtener una recta que mejor ajuste a un conjunto de puntos con tendencia lineal discriminando a los puntos atípicos. | 67 |
| 6.4 Anexo #4: Rutina en Matlab © para la agrupación de puntos en regiones homogéneas con el algoritmo k – means. | 68 |
| 6.5 Anexo #5: Rutinas diversas para la estandarización de perfiles. | 70 |
| REFERENCIAS | 78 |

CAPÍTULO 1. INTRODUCCIÓN.

1.1 INTRODUCCIÓN

Entre los factores críticos para el éxito de una industria de manufactura está el cumplimiento con los altos estándares de calidad. La inspección de piezas es una de las principales estrategias en el aseguramiento de la calidad en las industrias manufactureras al facilitar la reducción de fallas o inconsistencias en la materia prima, los productos en proceso y productos terminados. Durante la inspección, características de calidad se miden y se evalúan contra referencias para dictaminar la aceptabilidad del producto. En ocasiones, los tiempos de inspección son elevados debido, principalmente, a las precisiones requeridas y características morfológicas del producto en cuestión. Para reducir este tiempo se puede recurrir a la automatización, mediante la integración de sistemas electromecánicos con sensores de alta precisión, los cuales permitirían, con una mínima asistencia humana, dictaminar si las piezas a inspeccionar cumplen con las especificaciones de calidad o no.

1.2 ANTECEDENTES

La industria ha motivado la incorporación de sistemas y productos que ayuden al monitoreo y control de sus procesos. Entre estos destacan los sistemas de inspección asistidos por tecnología de visión auxiliada por operadores, y los sistemas autónomos capaces de detectar casi por sí mismos defectos morfológicos en los productos. Actualmente existen empresas que desarrollan tecnología para realizar inspecciones en este rubro, identificándose dos campos: (1) sistemas de visión por medio del reconocimiento de imágenes, y (2) sistemas de inspección por láser.

De estos últimos, se han presentados cuantiosos avances, a tal punto que la velocidad de sus mediciones y la precisión de las mismas los hacen atractivos para el desarrollo de sistemas más complejos de inspección en la producción de piezas con un relativo alto grado de confiabilidad.

A pesar de que los escáneres láser han sido ampliamente utilizados en años recientes para la inspección en industrias manufactureras, existen pocas investigaciones al respecto en nuestro país. En el extranjero está la publicación de Xi y Shu [1], donde muestran el desarrollo de un sistema de planificación de escaneo láser en 3D basado en gráficas CAD. Ellos probaron maximizar la cobertura de la pieza, encontrando la mejor forma de hacerlo para el campo de visión y orientación

del escáner laser. Sin embargo el sistema se enfocó en morfologías muy simples. Bernard y Véron [2] desarrollaron un nuevo método que automáticamente puede escanear objetos usando programación fuera de línea de máquinas de medición de coordenadas. Sin embargo la eficiencia del sistema requería ser mejorada para el escaneo de partes complejas. Zussman et al. [3,4] desarrollaron un algoritmo que determinó la localización del sensor láser. Pero este algoritmo solo se podía aplicar en un perfil 2D de una superficie y no podía ser utilizado para escanear la superficie completa de un objeto. Elber y Zussman [5], usando un método de descomposición de superficies, construyeron un algoritmo para calcular los números y direcciones óptimas de escaneo para superficies de cualquier forma. Sin embargo, ellos sólo consideraron el ángulo entre la superficie normal y el rayo de luz incidente para determinar si un punto de la superficie es factible para ser escaneado. No consideraron otras restricciones. Por otro lado, Yau y Menq [6] y Lim y Menq [7] desarrollaron un sistema automatizado de planeación en máquinas de medición de coordenadas para la inspección de partes complejas, donde la inspección de una parte manufacturada puede ser planeada en el ambiente CAD/CAM, y ejecutado por la CMM en el piso automáticamente. Spitz et al. [8] introdujeron nociones de accesibilidad y proximidad, y describieron dos tipos de algoritmos para acceder a la información en la computadora. Un algoritmo mejoró la computación exacta en un objeto poliédrico, aunque era relativamente lento, pero el uso de aproximación discreta logró incrementar su velocidad. El algoritmo discretizado fue probado en partes reales.

1.3 MOTIVACIÓN

La motivación para la realización del presente trabajo radica en que en la actualidad, la mayor parte de la inspección de piezas asistida por sensores láser se hace adquiriendo y procesando imágenes en dos dimensiones, sin embargo este enfoque presenta limitantes que un sistema capaz de reconocer tres dimensiones superaría en el análisis completo de la morfología de las piezas monitoreadas.

Por estas razones, en la presente investigación se desarrollaron estrategias matemáticas para la reconstrucción de imágenes en tres dimensiones partiendo de la información obtenida por escáneres láseres y sistemas electromecánicos, los cuales han sido proveídos a su vez por una empresa de integración de estas tecnologías. La calidad de la reconstrucción del modelo de la superficie de las piezas a inspeccionar, depende del tipo y precisión de los datos de los puntos medidos así como del equipo de medición [9, 10]. La integración de un equipo de inspección automatizado, que tenga alta confiabilidad en sus mediciones, así como rapidez en la inspección, lo vuelven una solución muy atractiva para las necesidades de inspección y aseguramiento de la calidad en las industrias manufactureras de hoy en día.

1.4 APORTACIONES

A continuación se mencionan las aportaciones que han surgido del desarrollo del presente trabajo.

1.4.1 Algoritmo para la reconstrucción 3D.

Se diseñó un algoritmo para la conversión y estandarización de los datos de las coordenadas de un perfil obtenido por el escáner, a un sistema de coordenadas global predeterminado, y que además es capaz de acoplar todos estos perfiles para generar un modelo 3D de una pieza de manufactura, la cual a la postre, se utiliza como modelo de comparación de la pieza contra un patrón preestablecido. Dicho algoritmo se presenta en el Anexo 2.

1.4.2 Algoritmos para la estandarización de perfiles.

Se desarrollaron algoritmos para la estandarización importante de los perfiles bidimensionales aportados por el escáner láser. Estos algoritmos se implementaron como rutinas en Matlab y su código se presenta en el Anexo 3.

Los resultados de la primera etapa de este trabajo se presentaron para su exposición oral en el 7o. Congreso Iberoamericano de Innovación Tecnológica y Desarrollo de Productos, celebrado en noviembre de 2012 en la ciudad de Orizaba, Veracruz.

1.5 ORGANIZACIÓN DE LA TESIS.

En el Capítulo 2 de esta obra, se presentan las características y funcionamiento del dispositivo láser y del sistema electromecánico empleados en la presente investigación. En el Capítulo 3 se da una explicación general de los conceptos matemáticos utilizados para desarrollar la modelación matemática requerida en el tratamiento de los datos. A saber, la utilización de los cuaterniones [11] para la conversión de coordenadas de puntos en el espacio como resultado de traslaciones y rotaciones, así como generalidades de muestreo de datos. Además es presentado el desarrollo de un algoritmo que permite discriminar puntos atípicos de un perfil así como el proceso de estandarización de los datos para la posterior reconstrucción tridimensional y comparación con modelos virtuales preestablecidos. En el Capítulo 4 se experimenta con los algoritmos desarrollados para comprobar la eficacia de los mismos. Registros de tiempo y confiabilidad son reportados. En el Capítulo 5 se analizan los resultados y se argumentan conclusiones. Recomendaciones para trabajos futuros relacionados son indicadas aquí también. Finalmente se presentan Anexos y Referencias.

CAPÍTULO 2. ASPECTOS TÉCNICOS DEL EQUIPO

2.1 ESCÁNER LÁSER

2.1.1 Aspectos básicos de sistemas de escaneo láser.

La configuración del dispositivo de escaneo láser, que se empleó en la presente investigación, se muestra en la Figura 1. Un haz láser es proyectado sobre la superficie a escanear formando una banda sobre esta. El haz reflejado es detectado por una cámara CCD.

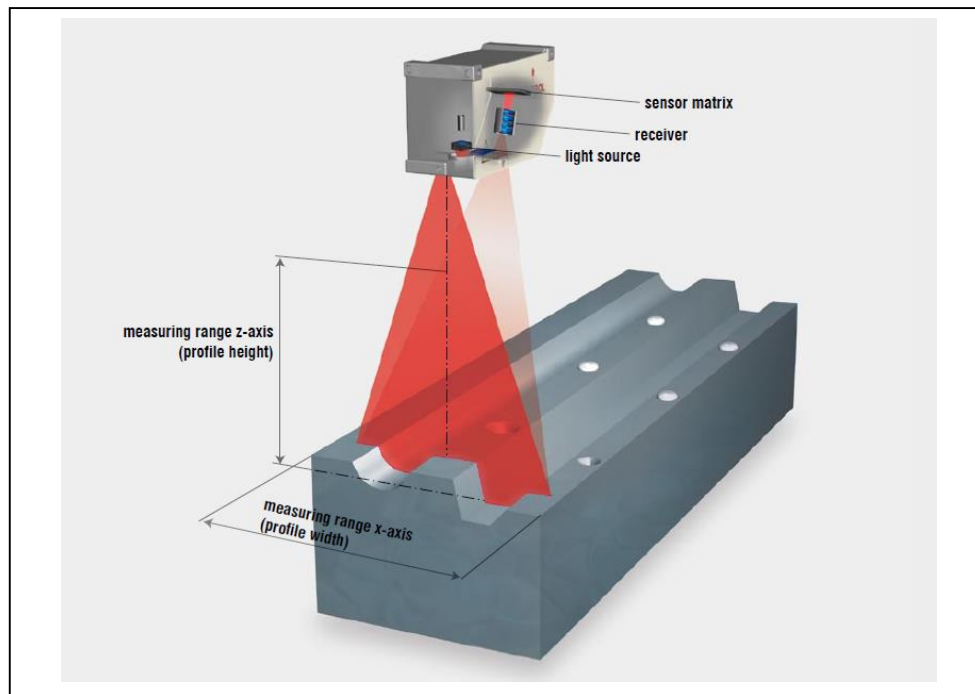


Figura 1. Esquema del funcionamiento del escáner láser.

Por medio del método de la triangulación, y por el procesamiento de imágenes descrito en el capítulo siguiente, se obtiene un conjunto de coordenadas tridimensionales, las cuales funcionan como un modelo 3D de la pieza que se inspecciona. El dispositivo láser es montado sobre un mecanismo electromecánico, que consiste en un brazo robótico en la presente investigación y que a su vez se describe en la sección siguiente. Dicho dispositivo electromecánico traslada al dispositivo láser a lo largo de una trayectoria pre – establecida de escaneo, la cual debe de consistir en una serie de caminos rectos, a altura constante. Las rotaciones en dos dimensiones también son permitidas. Cuando el escáner láser captura un perfil bidimensional, debe de mantenerse constante la altura del dispositivo (*stand off distance*) y por consiguiente, la longitud de la banda del haz láser proyectada sobre el objeto. Puesto que un escáner láser consiste de un conjunto de sensores ópticos y partes mecánicas móviles, se requiere satisfacer una serie de restricciones a la hora de procesar un punto en un objeto (Figura 2). A saber:

1. Ángulo de visión: el ángulo subtendido por el haz láser incidente y la normal a la superficie en el punto escaneado, debe ser menor que el máximo del ángulo de visión γ .
2. Campo de visión (*field of view*, FOD): El conjunto de puntos a medir debe de caer dentro de la banda láser proyectada sobre el objeto.
3. Profundidad de campo (*depth of field*, DOF): Los puntos medidos deben caer dentro de un rango de distancias especificadas de la fuente del haz láser.

4. Oclusión: El haz incidente, así como el haz reflejado no deben ser obstruidos con el objeto mismo.
5. El escáner láser debe de moverse en su trayectoria predefinida sin peligros de colisión.
6. Si existieran partes brillantes o transparentes en el objeto a inspeccionar, un pre – procesamiento como esprayado (*spraying*) puede ser requerido antes del escaneo láser para evitar pérdida de información o ruido indeseado.

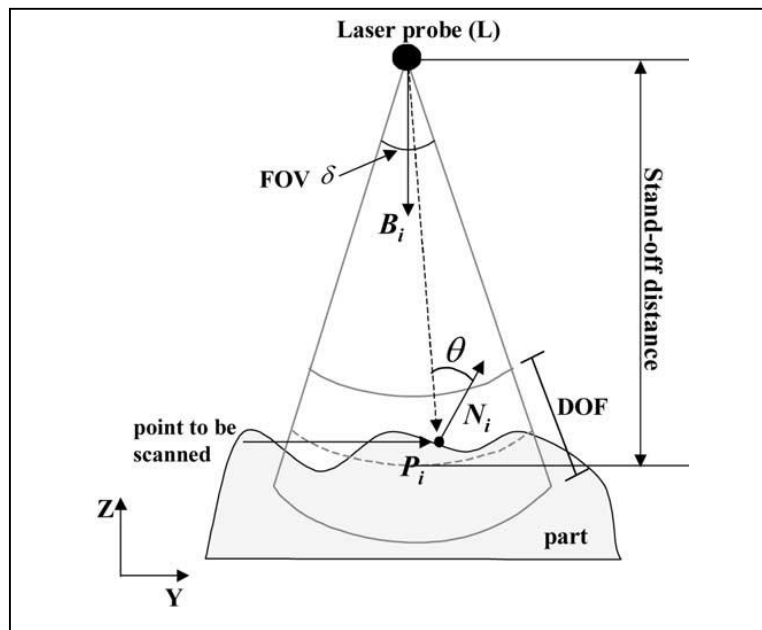


Figura 2: Esquema para el análisis de las restricciones en el escaneo láser.

2.1.2 Escáner Micro – Épsilon.

En la presente investigación, se trabajó con un escáner láser de alta precisión de la marca Micro-Épsilon modelo ScanControl 2810 - 100. El escáner es capaz de

obtener perfiles de una muestra con una resolución de 40 micrómetros y perfiles con hasta 1024 puntos. La velocidad en la adquisición de datos es de hasta 256 000 puntos/segundo [12]. Su fuente de luz es a base de láser semiconductor de 655 nm con una potencia de 15 mW. La forma en que opera así como el aspecto del mismo se muestra en la Figura 3.

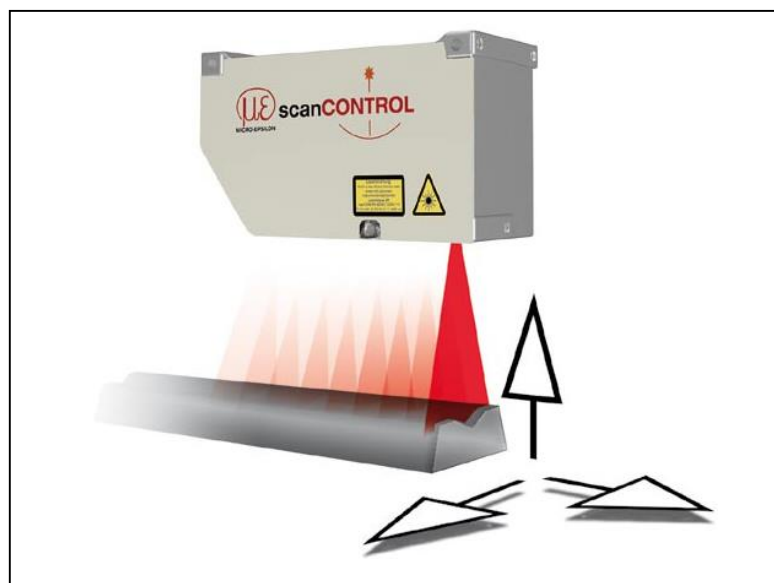


Figura 3. Escáner láser modelo Scan Control 2810 – 100.

2.2 DESCRIPCIÓN DEL SISTEMA ELECTROMECHANICO.

El sistema electromecánico que soporta al escáner láser en la presente investigación consiste en un brazo robótico de la marca ABB modelo IRB 1410. Su aspecto se muestra a continuación [13]:



Figura 4. Aspecto del brazo robótico ABB modelo IRB 1410

En la Tabla I se muestran algunas especificaciones del robot. [13]

| Tabla I. Especificaciones del brazo robótico modelo IRB1410 | | | | |
|---|--------------------|--------------------------|---------|-------------------------|
| Versión de Robot | Capacidad de carga | Alcance | Montaje | Opciones de protección. |
| IRB 1410 | 5 Kg | 1.44 m (en el 5º eje) | Suelo | Ambiente seco |

En la Figura 5 se muestra la región de alcance del robot en milímetros y el diagrama de carga.

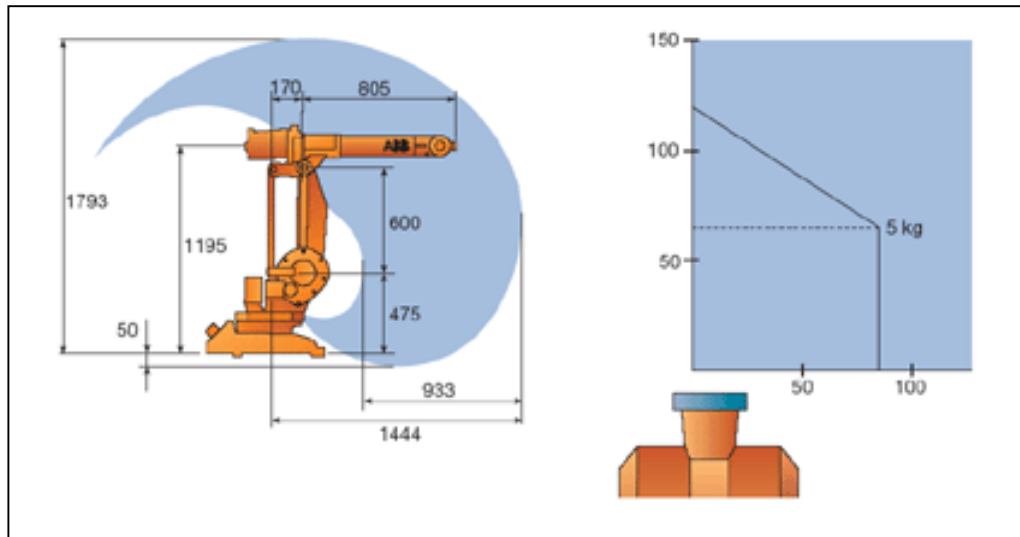


Figura 5. Alcance del brazo robot y diagrama de carga

Este robot tiene un peso de 225 kg. y cuenta con un total de 6 ejes. La velocidad máxima de movimiento es de 2.1 m/s y el fabricante lo produce especialmente para la industria de soldadura. La Figura 6 muestra una imagen de cómo queda el ensamble escáner – robot.



Figura 6. Ensemble escáner – robot.

CAPÍTULO 3. ANÁLISIS DE LA NUBE DE PUNTOS

3.1 PLANTEAMIENTO DEL PROCESO

El objetivo de la presente investigación consiste en analizar un conjunto de puntos bidimensionales proveídos por un escáner láser para discriminar sus puntos atípicos e identificar regiones rectilíneas homogéneas. Esto conformará un primer paso hacia la automatización de los sistemas de inspección con mínima asistencia humana. El proceso completo consta de tres fases:

1. Preparación previa al escaneo,
2. Escaneo y obtención de los datos y
3. Análisis y procesamiento de la información.

En la fase 1, se pretende que un plan de escaneo sea determinado por un grupo de ingenieros de cada planta, de acuerdo a los requerimientos preestablecidos y control de las restricciones. El número de barridos, las direcciones y trayectos son determinados en esta etapa. Cabe mencionar que no es objeto del

presente trabajo el determinar dicho plan. Se asume que se disponen de modelos CAD para cada pieza a inspeccionar los cuales se utilizan para definir dicho plan de escaneo.

En la fase 2, se debe primeramente posicionar a cada pieza en una posición preestablecida por medios manuales o bien por medios electromecánicos de manera precisa. En un trabajo futuro se pretende acondicionar dicho método para el caso en que las piezas a inspeccionar vayan entrando a la celda de inspección mediante una banda transportadora, y por ende, en una posición variable. A continuación, se escanea la pieza de acuerdo al plan de barrido preestablecido en la fase 1 y se registran los datos en un programa computacional apropiado para el procesamiento de dicha información. En la presente investigación, se utilizaron los programas Microsoft Excel® y MATLAB® de Mathworks. En esta etapa se construye un modelo tridimensional del tipo de nube de puntos de la pieza a inspeccionar, la cual se tomará como referencia para determinar, si dicha pieza está dentro de especificaciones de calidad o no. Dicho procedimiento se describe en el capítulo siguiente.

Finalmente en la etapa 3, se procesan los datos con varios objetivos: (a) Filtrar los datos a fin de excluir los puntos que estén fuera de la realidad del objeto y que son causados por ruido, así como completar las listas de puntos (perfiles), siguiendo la tendencia de estos, cuando estos se obtienen de manera incompleta por razones debidas al azar o a la variabilidad de los instrumentos de medición. (b) Muestrear los datos, sin perder la información relevante, para disminuir el costo

computacional (tiempo y uso de memoria computacional). Finalmente, con la nube de puntos estandarizada, se efectúa el análisis de los para determinar de manera objetiva si la pieza que se inspecciona está dentro de especificaciones o no.

En un trabajo futuro, se agregará un proceso para determinar además la localización física de las fallas en la pieza, y así, ahorrar más tiempo en el reprocesado de la pieza defectuosa. En la Tabla II se detallan las acciones del proceso completo.

| Tabla II. Etapas del proceso completo. | |
|---|--|
| Etapas | Acciones |
| 1. Generación del plan de escaneo | <ul style="list-style-type: none"> • Estimación de las trayectorias y direcciones de barrido óptimas. • Estimación del número de barridos a efectuar de acuerdo al tamaño de la pieza. • Verificación de que las restricciones ópticas como la oclusión y FOD se satisfagan. |
| 2. Ejecución del plan de escaneo. | <ul style="list-style-type: none"> • Posicionamiento correcto de la pieza a inspeccionar. • Escaneo de la pieza de acuerdo al plan. • Registro de los datos obtenidos en el escaneo en el software preestablecido. |
| 3. Análisis y Procesamiento de los datos | <ul style="list-style-type: none"> • Integración de los perfiles individuales en una nube de puntos tridimensional. • Filtrado de los datos para eliminar puntos atípicos aislados. • Estandarización de cada perfil individual para completar el número de puntos por perfil preestablecido cuando sea necesario. • Muestreo de los puntos que conforman a la nube de puntos para reducir costo computacional en los procesos siguientes. • Análisis para determinar si la pieza inspeccionada está dentro de especificaciones o no. |

En la presente investigación es deseable determinar la ubicación real de la pieza con respecto a un sistema de coordenadas previamente establecido, con la finalidad de tener una nube de puntos estandarizada que modele a la pieza para su posterior evaluación. Para esto es preciso analizar la relación de los desplazamientos y rotaciones del brazo robótico con las coordenadas del escáner láser. Una manera eficaz de describir dichas rotaciones, y que es ampliamente utilizada en mecatrónica, es mediante el uso de cuaterniones.

3.2 GENERALIDADES DE LOS CUATERNIONES

Los cuaterniones fueron desarrollados por Sir William Hamilton en 1843 [11] como una herramienta matemática con la que fuera posible describir el espacio y tiempo de una forma precisa. Los cuaterniones son una extensión de los números reales similar a los números complejos, en los que en lugar de añadir solamente al número imaginario i , se añaden tres. A saber i , j y k , con la propiedad tal que $i^2 = j^2 = k^2 = ijk = -1$. De esta manera, se define al cuaternión q como el número con la forma $q = q_1 + q_2i + q_3j + q_4k$, donde $q_1, q_2, q_3, q_4 \in \mathbb{R}$. A la componente q_1 se le denomina *parte escalar* del cuaternión mientras que a la suma $q_2i + q_3j + q_4k$ se le llama *parte vectorial* del cuaternión. Un cuaternión unitario \hat{q} es aquel cuya norma euclidiana es igual a uno:

$$\hat{q} = q_1 + q_2\hat{i} + q_3\hat{j} + q_4\hat{k} \quad (1)$$

$$\|\hat{q}\|_2 = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \quad (2)$$

La importancia de los cuaterniones unitarios en el presente trabajo radica en que con ellos se pueden representar rotaciones alrededor de un eje en el espacio de una manera sencilla.

Proposición 1. Si un punto P tiene coordenadas $P = (x_1, y_1, z_1)$ con respecto a un cierto sistema de coordenadas y luego dicho sistema se rota θ radianes en dirección horaria con respecto a un cierto eje con la dirección del vector unitario \hat{v} , entonces, su nueva posición, P' , viene dada por:

$$P' = \text{vec}\{qP\bar{q}\} \quad (3)$$

con

$$q = q_1 + q_2\hat{i} + q_3\hat{j} + q_4\hat{k} = \cos\left(\frac{\theta}{2}\right) + \hat{v}\text{sen}\left(\frac{\theta}{2}\right) \quad (4)$$

■

En la ecuación (3), el operador $\text{vec}\{ \cdot \}$ devuelve la parte vectorial del cuaternión q . Es decir, $\text{vec}\{q_1, q_2, q_3, q_4\} = (q_2, q_3, q_4)$. Al cuaternión \bar{q} se le denomina cuaternión conjugado de q , cuya forma explícita es $\bar{q} = q_1 - q_2\hat{i} - q_3\hat{j} - q_4\hat{k}$. Además, cabe señalar que el producto indicado en (3) no es conmutativo, esto es $qP\bar{q} \neq \bar{q}Pq$ (el orden de los factores si altera al producto). La Tabla III, muestra algunas propiedades de los productos de los números imaginarios i, j y k .

| Tabla III. Productos de los números cuaterniónicos i, j y k . | | |
|---|------------|------------|
| $i^2 = -1$ | $j^2 = -1$ | $k^2 = -1$ |
| $ij = k$ | $jk = i$ | $ki = j$ |
| $ik = -j$ | $kj = -i$ | $ji = -k$ |

Ejemplo 1: Una aplicación se coloca en la muñeca de un brazo robótico (triángulo rojo) como se muestra en la Figura 7.

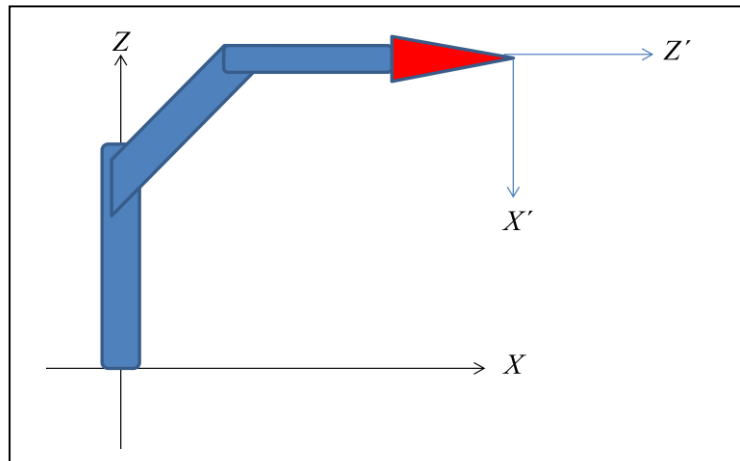


Figura 7. Esquema del ejemplo 1. Los ejes X' y Z' indican la posición del sistema de coordenadas después de la rotación.

Suponga que inicialmente la punta de la aplicación posee las coordenadas $P(1,2,3)$ con respecto a la base del robot. A continuación, la aplicación en la muñeca del robot se rota 90° en sentido horario con respecto al eje y (que sale de la hoja). La pregunta es ¿cuáles son las nuevas coordenadas de la punta de la aplicación

después de efectuada la rotación? Para contestar esta pregunta, se empleará la ecuación (3).

Primeramente, obtenemos el cuaternión q que indica la rotación en el eje indicado. De la ecuación (4) tenemos:

$$\begin{aligned} q &= \cos\left(\frac{90^\circ}{2}\right) + (0,1,0) \operatorname{sen}\left(\frac{90^\circ}{2}\right) \\ q &= \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}(0,1,0) \\ q &= \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \hat{j} \end{aligned}$$

A continuación, empleamos directamente la ecuación (3), además de la propiedad distributiva de la multiplicación y las propiedades de la Tabla IV para obtener las nuevas coordenadas del punto P :

$$\begin{aligned} P' &= qP\bar{q} \\ &= \left(\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j\right)(i+2j+3k)\left(\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j\right) \\ &= \left(-\sqrt{2} + 2\sqrt{2}i + \sqrt{2}j + \sqrt{2}k\right)\left(\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j\right) \\ &= 3i + 2j - k \end{aligned}$$

Por lo que las nuevas coordenadas del punto P , P' , son $P'(3,2,-1)$. ■

3.3 RELACIÓN ENTRE LAS COORDENADAS DEL ESCÁNER LÁSER Y EL SISTEMA DE COORDENADAS GLOBAL.

El escáner láser proporciona un perfil bidimensional, con coordenadas X' y Z' , en milímetros, de la morfología de una pieza de manufactura como se muestra en la Figura 8.

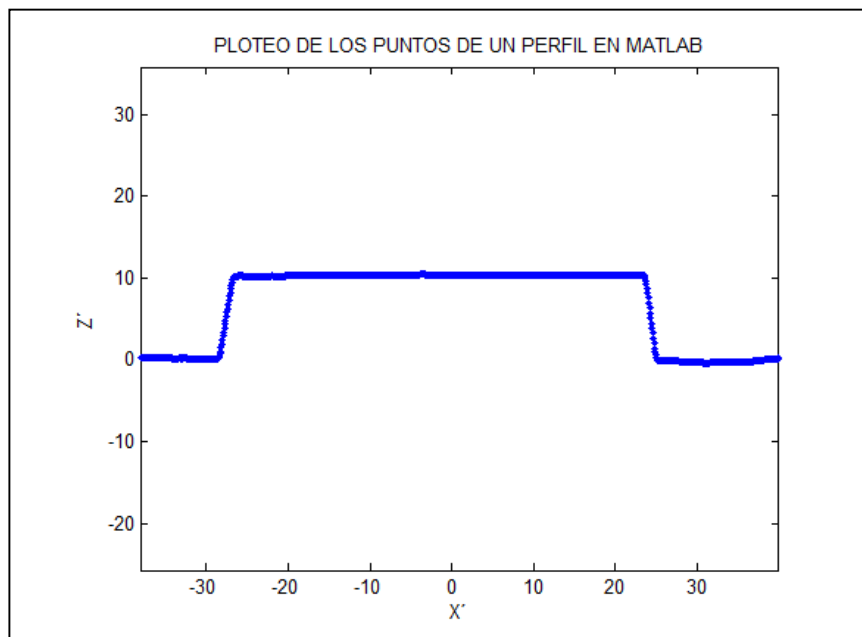


Figura 8. Gráfica de un perfil bidimensional real proporcionado por el escáner láser de una pieza metálica de sección rectangular de 10x50 mm.

El perfil mostrado en la Figura 8 es el caso ideal en el que se registran 1024 puntos por perfil en total, es decir el número de puntos preestablecido por el escáner por perfil. Sin embargo, hay casos en los que el número de puntos obtenidos por perfil es menor que 1024, como el de la Figura 9.

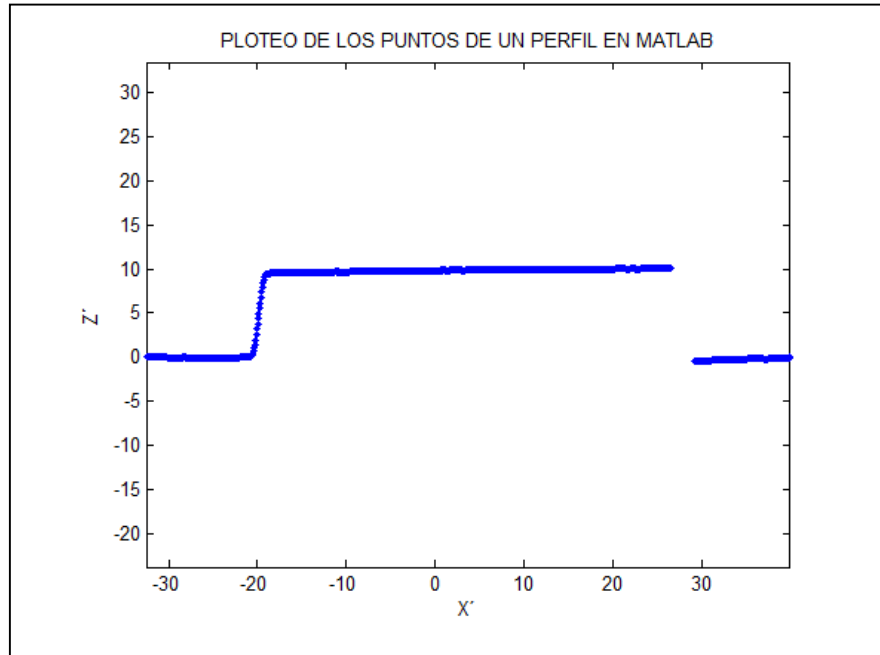


Figura 9. Perfil obtenido para la misma pieza que el de la Figura 8, pero con 894 puntos en lugar de los 1024.

La diferencia se debe al tiempo de exposición del haz láser y del ángulo con el que fueron obtenidos los datos. En la prueba que se registra en la Figura 9, el ángulo al que se dispuso el escáner láser fue de 45° con respecto a la normal de la superficie escaneada, mientras que en la prueba de la Figura 8, el ángulo fue de 90° .

Para los propósitos de la presente investigación, es deseable transformar estas coordenadas en el sistema de referencia del escáner láser a un sistema de coordenadas global (SCG) tridimensional, agregando la tercer componente (Y) y redefiniendo el origen de coordenadas en la base del robot, no en el sistema de

coordenadas del escáner. El diagrama del ensamble escáner – robot y su relación entre coordenadas se ilustra en la Figura 10.

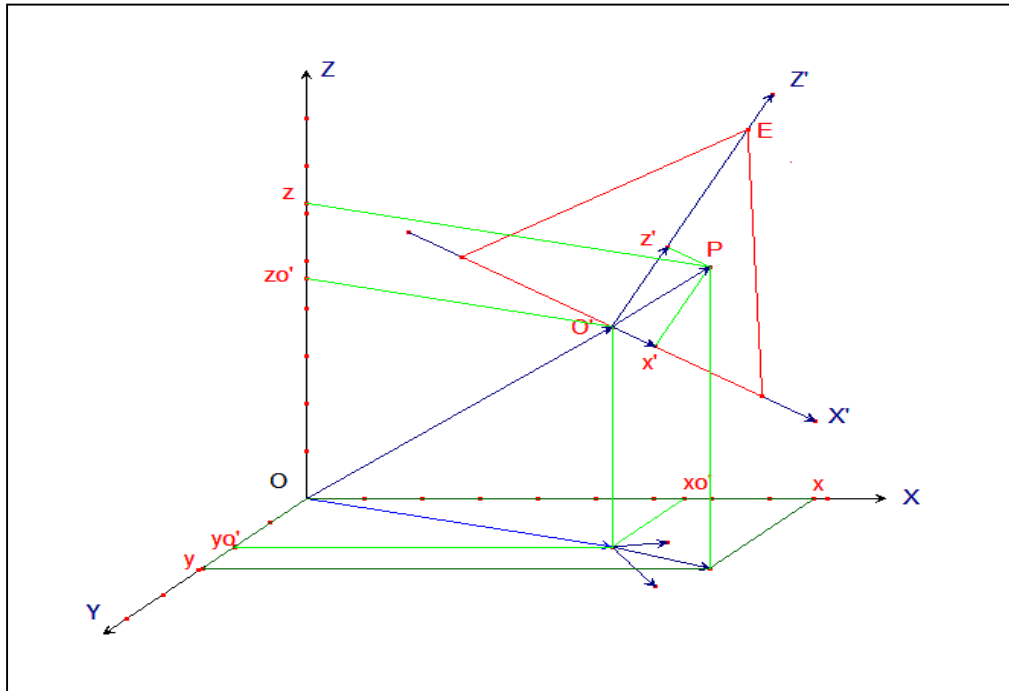


Figura 10. Relación entre las coordenadas de un punto P en el sistema de referencia del escáner (2D) y en un sistema de referencia global (3D)

Donde el vector $\overrightarrow{OO'}$ indica las coordenadas del muñón del robot con respecto a la base del robot (origen del SCG), lugar donde se coloca el escáner láser. De manera similar, el vector $\overrightarrow{O'P}$ indica la posición del un punto P en un perfil, pero con respecto a las coordenadas del escáner. Esto es, el punto P tiene coordenadas (x', z') . Ahora se deducirán las coordenadas de este mismo punto P pero con respecto al SCG, considerando las siguientes suposiciones.

1. Sistema de coordenadas global (SCG) fijo en la base del robot. Aunque puede definirse el origen de coordenadas en otro lugar.
2. Se conoce, a partir de la información del robot, las coordenadas del escáner con respecto al sistema de coordenadas global al tiempo discreto t .
3. Se conoce la orientación el plano $X'Z'$ con respecto al sistema de referencia global. Esto es, se conocen un vector unitario $\hat{u}_{X'}$ en la dirección X' y un vector unitario $\hat{u}_{Z'}$ en la dirección Z' ambos al tiempo discreto t .
4. El conjunto de puntos de cada perfil al tiempo t dado por el escáner. Esto es, un conjunto de hasta 1024 puntos en coordenadas X' y Z' .

De la Figura 10 se observa que las coordenadas (x,y,z) de un punto P en el perfil puede obtenerse a partir de la suma de vectores siguiente:

$$(x, y, z) = \overrightarrow{OO'} + \overrightarrow{O'P} \quad (5)$$

Donde a su vez

$$\overrightarrow{OO'} = (x_E, y_E, z_E) \quad (6)$$

$$\overrightarrow{O'P} = x' \hat{u}_{X'} + z' \hat{u}_{Z'} \quad (7)$$

Siendo x' y z' las coordenadas del punto P del perfil dadas por el escáner y $\hat{u}_{X'}$ y $\hat{u}_{Z'}$ vectores unitarios en la dirección de X' y Z' respectivamente con respecto

al SCG. Así mismo las coordenadas del escáner, al tiempo t , vienen indicadas por (x_E, y_E, z_E) , información proporcionada por el sistema del robot. La orientación del plano $X'Z'$ al tiempo t , también es indicada por el sistema electromecánico mediante las componentes de un cuaternión unitario.

De esta manera, los vectores unitarios $\hat{u}_{X'}$ y $\hat{u}_{Z'}$, pueden ser obtenidos a partir de (3) y (4) de la forma siguiente:

$$\hat{u}_{X'} = \text{vec}\{q(0,1,0,0)\bar{q}\} \quad (8)$$

$$\hat{u}_{Z'} = \text{vec}\{q(0,0,0,1)\bar{q}\} \quad (9)$$

Donde $\text{vec}\{Q\}$ representa la parte vectorial del cuaternión Q . Observe que $(0,1,0,0)$ es el cuaternión unitario en la dirección X , y similarmente $(0,0,0,1)$ el cuaternión unitario en la dirección Z (del SCG).

Proposición 2. Sean $P(x_1', z_1')$ las coordenadas de un punto cualquiera del perfil en el sistema de coordenadas del escáner láser. Sean además (x_E, y_E, z_E) las coordenadas del escáner láser al momento en que se obtuvieron las coordenadas de P , en el sistema de coordenadas global (SCG), con la dirección indicada por el cuaternión q . Entonces, las coordenadas del punto P , $P(x, y, z)$, con respecto al SCG vienen dadas por:

$$P(x, y, z) = (x_E, y_E, z_E) + x_1' \cdot \hat{u}_{X'} + z_1' \cdot \hat{u}_{Z'}$$

siendo $\hat{u}_{X'}$ y $\hat{u}_{Z'}$, como en las ecuaciones (8) y (9) respectivamente.

■

La prueba de la proposición 2 se obtiene a través de una simple suma de vectores.

La idea principal es reconstruir una superficie 3D obteniendo el i -ésimo perfil $X_i - Z_i$ e incorporándole la correspondiente componente Y_i . Al correr los índices desde 1 hasta n , se obtendrá el resultado deseado (Figura 11)

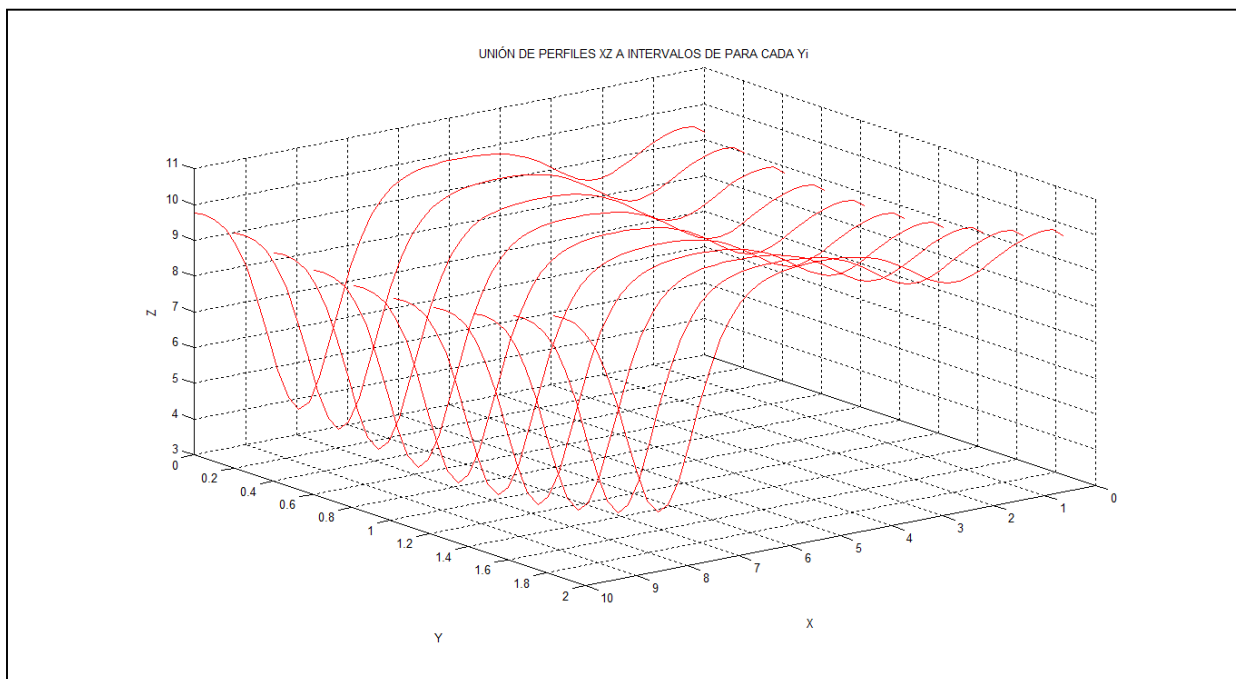


Figura 11. Unión de n perfiles para formar una malla 3D que a su vez modela una superficie

Sin embargo, si las piezas a inspeccionar son más grandes que lo que la línea del haz láser puede abarcar, entonces se tiene la necesidad no sólo de obtener una sola malla tridimensional, sino de tomar lecturas de cada una de las cinco caras vistas de un paralelepípedo orientadas en direcciones distintas. (Figura 12).

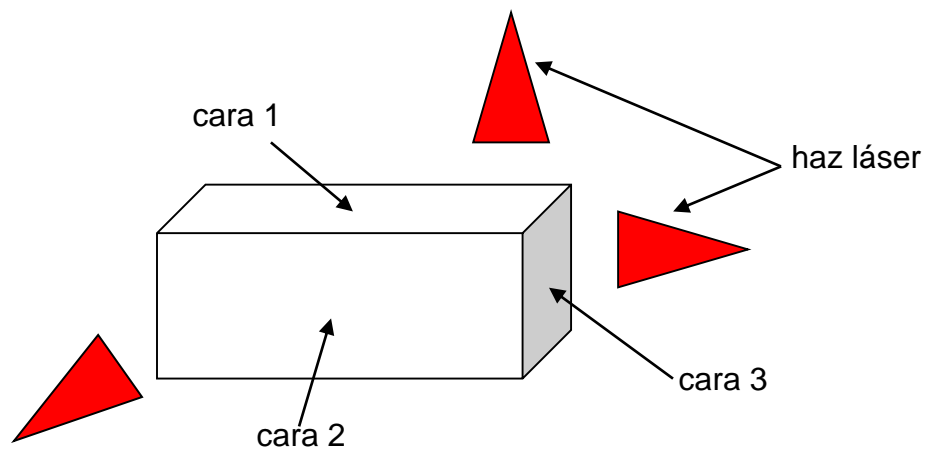


Figura 12. Obtención de la morfología de cada una de las caras de una pieza grande.

3.4 ALGORITMO PARA LA CONSTRUCCIÓN DE UN SÓLIDO GEOMÉTRICO A PARTIR DE LA INTEGRACIÓN DE PERFILES BIDIMENSIONALES.

Se diseñó un algoritmo para integrar perfiles reales aportados por el escáner láser en una malla tridimensional descrita en términos del sistema de coordenadas global (SCG) y posteriormente se implementó en Matlab (Anexo 2).

Algoritmo #1: Conversión de los puntos en el sistema coordenadas del escáner al sistema de coordenadas global

1. Leer el conjunto de puntos M aportados por el escáner láser. M es una matriz de $m \times 2055$ elementos, siendo m el número de disparos hechos a distintos valores de la coordenada y . La primera columna de M representa el vector X_E , la segunda columna, al vector Y_E , la tercera columna de M , representa al vector Z_E , siendo estos vectores los que indican la posición del escáner láser al m – ésimo disparo. Cada uno de estos vectores, X_E , Y_E y Z_E son de tamaño $m \times 1$. Las siguientes 4 columnas representan las componentes del m – ésimo cuaternión Q : Q_1 es la cuarta columna de M , Q_2 la quinta, Q_3 la sexta y Q_4 la séptima columna de M . Siendo Q_1 el vector cuyos elementos conforman a la primera componente del m – ésimo cuaternión Q , Q_2 el vector cuyos elementos conforman a la segunda componente del m – ésimo cuaternión Q , Q_3 es el vector cuyos elementos constituyen la tercera componente del m – ésimo cuaternión Q y Q_4 es el vector

cuyos elementos constituyen a la cuarta componente del m – ésimo cuaternión Q . Cada una de estas matrices Q_1 , Q_2 , Q_3 y Q_4 son de tamaño $m \times 1$. Recuerde el lector que el cuaternión Q indica la orientación espacial a la cual fueron obtenidos los datos del escáner. La parte restante de la matriz M constituyen los conjuntos de datos aportados por el escáner, es decir, las matrices X' y Z' que describen a los perfiles bidimensionales. Cada una de estas matrices es de tamaño $m \times 1024$.

2. Inicio de un ciclo *for*. Los pasos del 3 al 8 se efectúan desde $i = 1$ hasta $i = m$ veces, es decir, para cada disparo o incremento en la coordenada y .

3. Obtener el cuaternión conjugado Q_c al i – ésimo cuaternión Q .

4. Formar la matriz de coordenadas del escáner C_E mediante la operación

$$C_E(i) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{1024 \times 1} (x_E(i) \ y_E(i) \ z_E(i))_{1 \times 3}$$

aquí las coordenadas (x_E, y_E, z_E) representan las coordenadas del escáner al i – ésimo disparo o incremento en y . Note que esta matriz C_E tiene tamaño 1024×3 .

5. Efectuar los productos cuaterniónicos U_x y U_z :

$$\begin{aligned} U_x(i) &= Q \langle 0, 1, 0, 0 \rangle Q_c \\ &= \langle Q_1(i), Q_2(i), Q_3(i), Q_4(i) \rangle \cdot \langle 0, 1, 0, 0 \rangle \cdot \langle Q_1(i), -Q_2(i), -Q_3(i), -Q_4(i) \rangle \\ U_z(i) &= Q \langle 0, 0, 0, 1 \rangle Q_c \\ &= \langle Q_1(i), Q_2(i), Q_3(i), Q_4(i) \rangle \cdot \langle 0, 0, 0, 1 \rangle \cdot \langle Q_1(i), -Q_2(i), -Q_3(i), -Q_4(i) \rangle \end{aligned}$$

6. Tomar la parte vectorial de cada cuaternión U_x y U_z para formar

los vectores unitarios \hat{u}_x y \hat{u}_z en las direcciones x y z respectivamente del Sistema de Coordenadas Global.

$$\hat{u}_x = \text{vec}\{U_x\}$$

$$\hat{u}_z = \text{vec}\{U_z\}$$

7. Inicio de otro ciclo for (anidado). La operación de multiplicación siguiente se efectúa 1024 veces (para cada punto en el m – ésimo perfil dado por el escáner láser).

$$X = x' \cdot \hat{u}_x$$

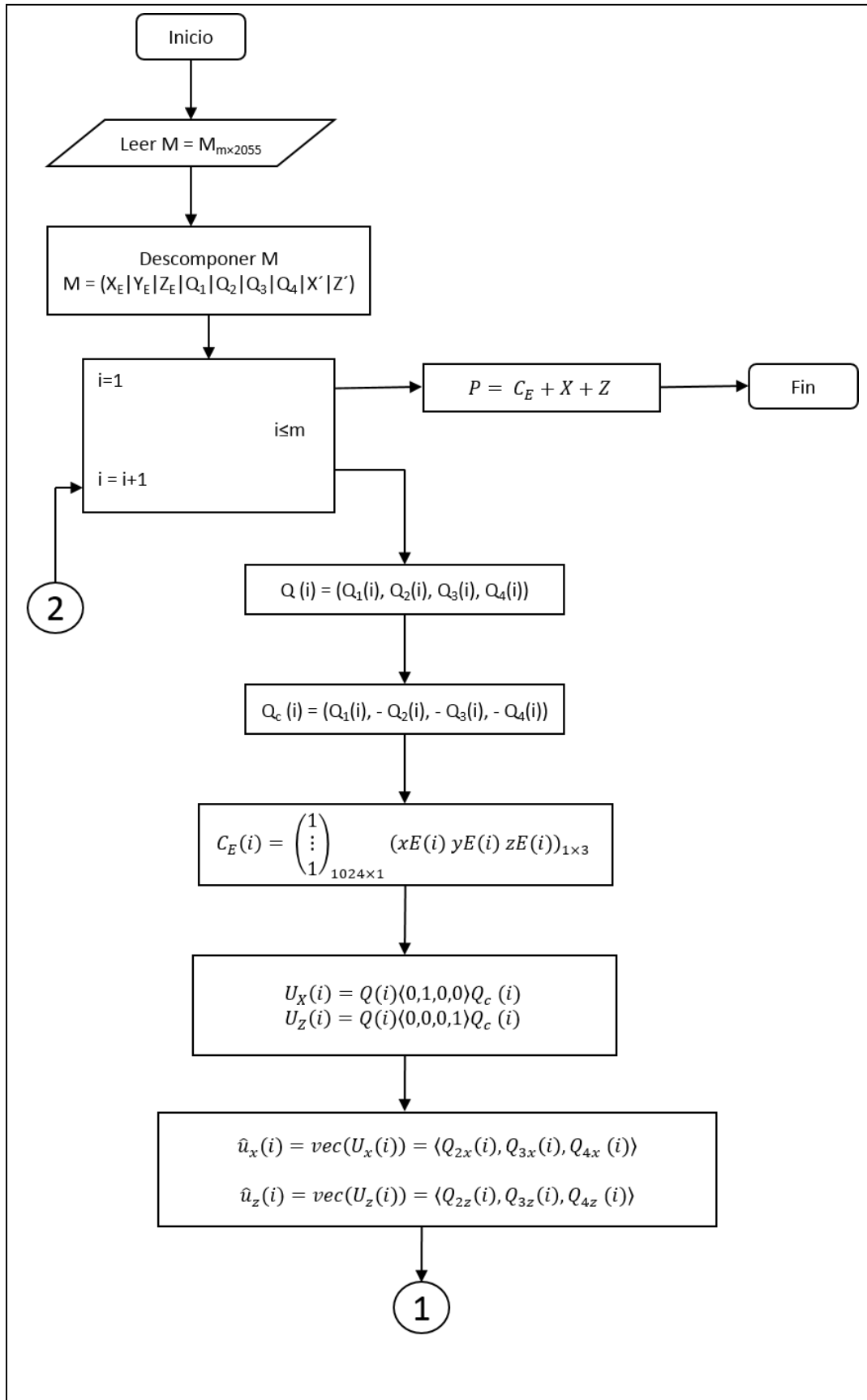
$$Z = z' \cdot \hat{u}_z$$

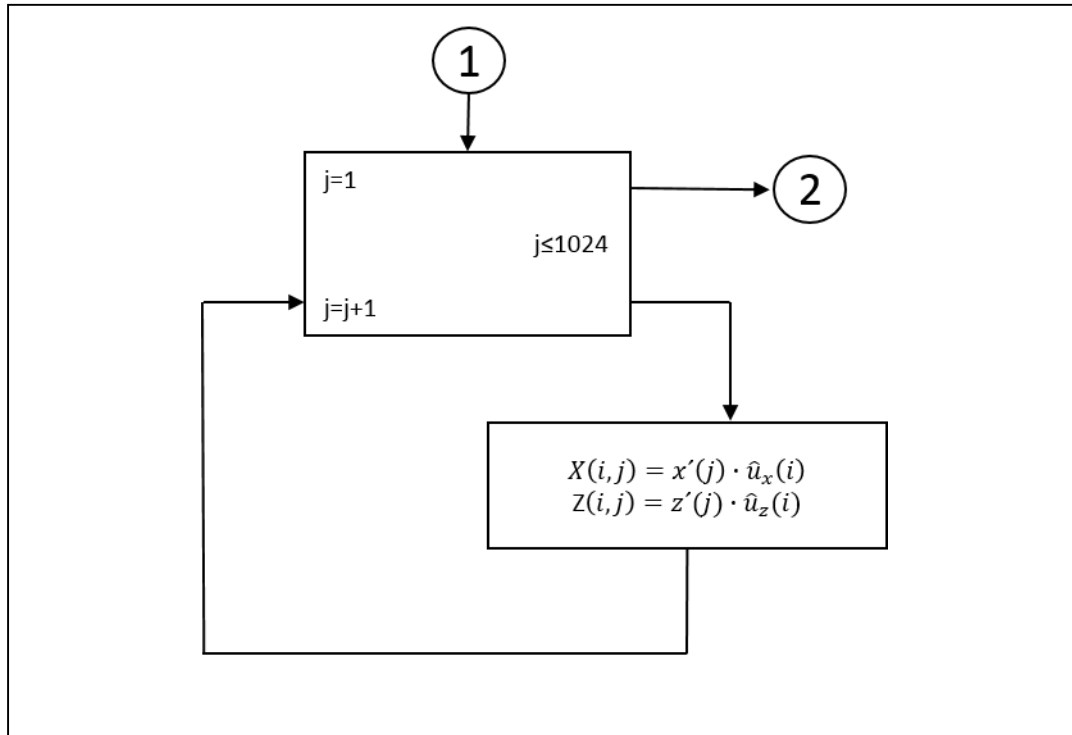
al término de las 1024 multiplicaciones, las matrices X y Z tendrán un tamaño de 1024×3 cada una.

8. Efectuar la suma matricial $C_E + X + Z = P$. La matriz P así formada, la cual es de tamaño 1024×3, representa al conjunto de puntos de un perfil bidimensional en las coordenadas del escáner escritas ya en términos del SCG.

9. Volver al paso 2, si aún no se completan los m ciclos. De lo contrario, terminar.

A continuación se describe el algoritmo en un diagrama de flujo.





En el Capítulo 4 Experimentación, se muestran los resultados después de haber implementado este algoritmo en Matlab.

3.5 ANÁLISIS Y PROCESAMIENTO DE DATOS.

Una vez que se ha concluido la etapa de la adquisición de datos, lo que sigue es procesar la información para su posterior análisis. En la etapa anterior, generalmente se obtiene una gran cantidad de datos denominada *nube de puntos*, la cual describe la geometría y morfología de un objeto real. Esta nube de puntos puede ser interpretada y analizada con diferentes programas tales como MatLab o LabView. Sin embargo, procesar la información tal y como se entrega por el escáner conlleva una alta carga computacional. Gracias a los avances en las tecnologías de adquisición de datos, particularmente en los escáneres láser, se pueden realizar muestreos de alta densidad sobre la superficie de objetos 3D, muestreos que pueden entregar millones de puntos para representar la morfología de un objeto. La reducción de la cantidad de las muestras tomadas es la clave de las técnicas de procesamiento para el diseño de algoritmos de visualización y de modelación.

A continuación se describe un método de simplificación de nube de puntos, el cual no requiere la construcción previa de mallas poligonales y es robusto al ruido y a valores atípicos presentes en los datos. Este método propone de tres etapas. En la primera se segmenta la nube de puntos en regiones homogéneas empleando el conocido algoritmo k-means. En la segunda etapa, se ajusta una línea de tendencia robusta al ruido en cada clúster para determinar la tendencia local de los puntos, mediante una variante del análisis de componentes principales (WPCA). Finalmente, en la tercera etapa, se efectúa la reducción de puntos en cada clúster basándonos en los puntos que posean una ponderación local mayor.

3.5.1 Agrupamiento o clusterización por medio del algoritmo *k-means*.

El agrupamiento tiene como objetivo determinar regiones de puntos homogéneas. En estas regiones se puede efectuar la reducción de puntos pero sin la pérdida de información significativa. El proceso para realizar la agrupación se lleva a cabo con el algoritmo k-means. La aplicación del k-means encuentra grupos o regiones con características homogéneas. El algoritmo *k-means* (estándar) se puede ejecutar en 4 fases [15]:

1. **Inicialización:** Se definen un conjunto de objetos a particionar, el número de grupos (k) y el centroide (media ponderada) para cada grupo. Algunas implementaciones del algoritmo estándar determinan centroides iniciales de manera aleatoria, mientras que otros procesan los datos y determinan los centroides mediante cálculos.
2. **Clasificación.** Para cada dato de la nube de puntos, se calcula su distancia a cada centroide y se determina el centroide más cercano, para incorporar a dicho valor al grupo con este centroide.
3. **Cálculo de centroides:** Para cada grupo generado en el paso anterior se vuelve a calcular el centroide.
4. **Condición de convergencia:** Se pueden ejecutar iteraciones entre los pasos 2 y 3 de acuerdo a un plan preestablecido o también es común parar las iteraciones cuando no existan intercambios de los datos entre los grupos. Así mismo, otro criterio de paro consiste en declarar la convergencia cuando

la diferencia entre los centroides de dos iteraciones consecutivas es menor que una tolerancia preestablecida.

Cabe señalar que este algoritmo heurístico presenta los siguientes inconvenientes:

- El agrupamiento final depende de los centroides iniciales. Los cuales son arbitrarios en primera instancia.
- La convergencia en el óptimo global no está garantizada y en problemas con muchos datos, requiere de una cantidad considerable de iteraciones y por ende, de carga computacional.

En la Figura 13 se muestra cómo resulta una imagen después de aplicar el algoritmo.

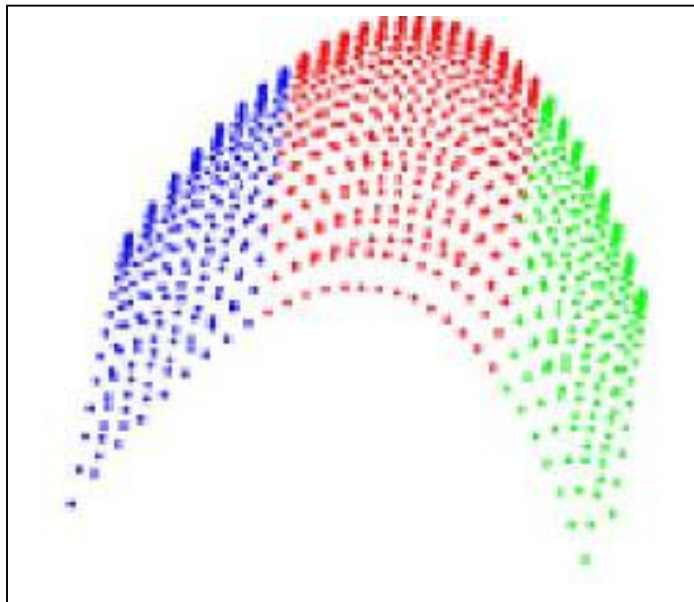


Figura 13. Regiones homogéneas encontradas por el k-means ($k=3$)

3.5.2 Análisis de Componentes Principales y una variante ponderada.

El análisis de componentes principales (ACP) es un método estadístico que describe la covarianza de una serie de datos en términos de algunas componentes expresadas como combinaciones lineales de las variables originales (Hubert, 2005). Es decir, el ACP reduce la dimensionalidad de un conjunto de datos perdiendo la menor cantidad de información posible. Las nuevas componentes, además, son independientes entre sí. Cabe destacar que un análisis de componentes principales tiene sentido si existen altas correlaciones entre las variables, ya que esto es indicativo de que existe información redundante, y por lo tanto, pocos factores representarán gran parte de la variabilidad total.

La primera componente del ACP corresponde a la dirección en la cual el conjunto de datos proyectados tienen la más alta varianza. La segunda componente, la cual es ortogonal a la primera, maximiza la varianza de los puntos proyectados sobre ella. El análisis de componentes principales es ampliamente utilizado en la identificación de sesgo de conjunto de datos. Se emplea en la reducción de dimensionalidad y visualización (Jolliffe, 1986), agrupamiento de datos (Pauly, 2002) y reconocimiento de patrones (De la Torre, 2001). A pesar de la versatilidad del ACP, esta técnica es sensible a puntos atípicos en los datos, es decir, puntos que se salen de la tendencia del resto de los puntos. En la Figura 14 (a), se observa que la mayoría de los puntos se agrupan en la parte baja de la figura. Tres de ellos, circulados en rojo, son puntos atípicos (outliers). La primera componente, mostrada como una línea azul, indica la dirección principal, como una

recta de regresión, de la dispersión de los puntos, pero se observa un sesgo debido a la influencia de los *outliers*. En la Figura 14 (b) se muestra la corrección con una variante robusta del ACP.

En este método, se toma una vecindad $N(p_i)$ alrededor del i – ésimo punto p_i . Enseguida, se calcula la media \bar{p}_i , con la ecuación 10

$$\bar{p}_i = \frac{1}{n} \sum_{j=1}^n p_j, \quad p_j \in N(p_i) \quad (10)$$

finalmente, usando la ecuación 11 se obtiene la matriz de covarianzas MC:

$$MC = \frac{1}{n} \sum_{j=1}^n (p_j - \bar{p}_i)(p_j - \bar{p}_i)^T \quad (11)$$

Se observa que la matriz de covarianzas MC es una matriz simétrica semi – definida positiva de tamaño 3×3 (ya que $p_i \in \mathbb{R}^3$). La descomposición en valores propios de la MC produce las componentes principales con sus asociados tres vectores propios v_1, v_2, v_3 y sus tres valores propios reales $\lambda_1, \lambda_2, \lambda_3$ correspondientes. Los eigenvalores miden la variación de los puntos en una vecindad a lo largo de las direcciones indicadas por sus correspondientes eigenvectores. Los eigenvectores ortogonales v_2, v_3 definen las direcciones de la más alta variación y su proyección resulta en un plano de ajuste en una vecindad. El eigenvector v_1 aproxima un vector normal en el p_i .

3.5.3 Estimación de la tendencia de los puntos en cada grupo.

Una vez establecidos los grupos, se determinan las tendencias locales de los puntos sobre la superficie, empleando una variante ponderada del análisis de componentes principales (WPCA), la cual asigna factores de peso (ponderaciones) a los datos mediante repartición de pesos inversamente proporcional a la distancia media del centro del grupo. Esta variante no requiere parámetros externos asignados por el usuario, ya que los pesos son calculados a partir de los datos mismos.

La propuesta de cálculo de pesos consiste en realizar una repartición de la suma total de las distancias de los puntos a la media, inversamente proporcional a la distancia a que los puntos se encuentran de ésta, es decir, los puntos más lejanos tendrán menos influencia sobre el plano (o línea) de regresión que los puntos más cercanos. Los factores de peso (w_i) se calculan con la ecuación 10:

$$w_i = \frac{1}{d_i \cdot \sum_{j=1}^n \frac{1}{d_j}} \quad (12)$$

con la propiedad

$$\sum_{i=1}^n w_i = 1 \quad (13)$$

En la ecuación 12, d_i es la distancia del i -ésimo punto a la media (punto medio) y n es el número de puntos que participan en el cálculo del plano de regresión. El empleo del análisis de componentes principales ponderado (WPCA) hace de este método robusto al ruido y a los valores atípicos. En la Figura 14 se aprecia la influencia indeseable de los puntos atípicos, circulados en rojo antes y después de aplicar el WPCA:

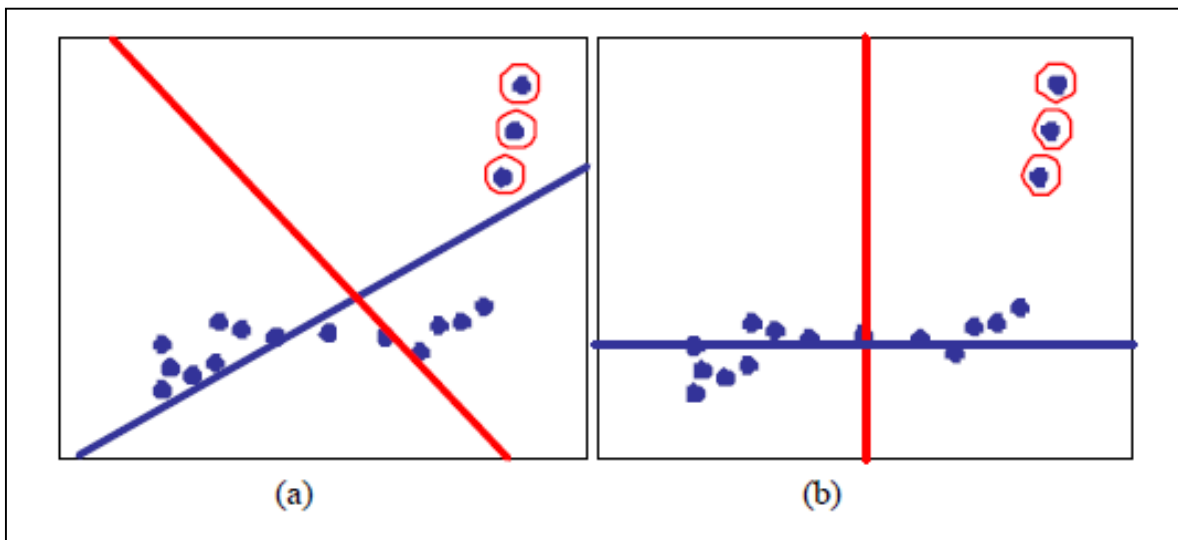


Figura 14. Análisis de componentes principales. (a) Componentes principales de los datos influenciados por valores atípicos. (b) Corrección de la tendencia generada por los valores atípicos, luego de aplicar a los puntos los factores de peso dados por la ecuación 10.

La línea azul ajusta mejor a los puntos, en el sentido que discrimina a los puntos atípicos y minimiza la suma de los errores, que por ejemplo, utilizar el enfoque simple de mínimos cuadrados (en rojo). Sin embargo, para que la modelización con la ecuación 12 sea efectiva, es preciso que el conjunto de puntos (x,y) sean trasladados al eje x . Es decir, si la ecuación $y = mx + b$ es una línea de

tendencia del conjunto de puntos, entonces antes de aplicar la ecuación 10 es necesario trasladar los puntos a cero mediante la transformación:

$$y_i^* = y_i - mx_i - b$$

La Figura 15 muestra la tendencia local de los datos dada por los planos WPCA calculados con el uso de la ecuación 12. Se aprecia la homogeneidad de los grupos de datos. Gracias a esto, la reducción realizada en cada uno de ellos no ocasiona pérdida de información representativa.

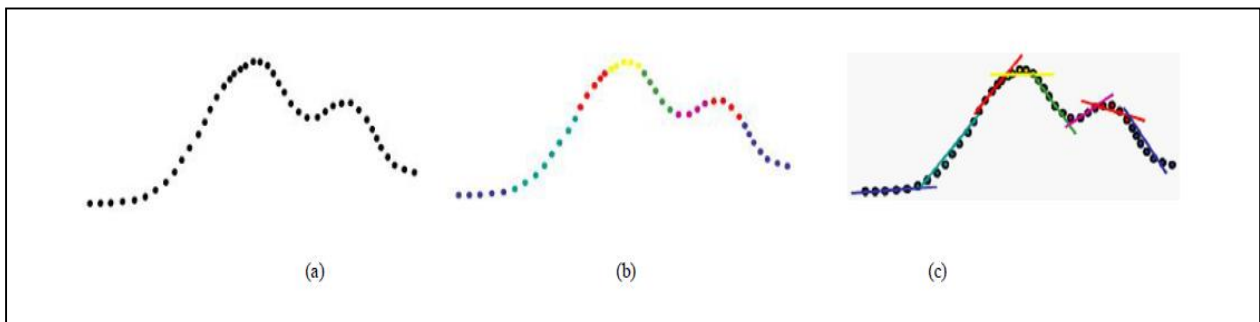


Figura 15. Análisis de la tendencia de los datos. (a) Nube de puntos (extracto). (b) Datos agrupados con k-means. (c) Planos de ajuste que describen la tendencia local de los puntos.

3.5.4 Reducción de la nube de puntos.

Una vez que se ha establecido la tendencia local de los puntos con el método de análisis de componentes principales ponderado, lo que sigue es reducir la cantidad de puntos de la nube, para su mejor procesamiento computacional y análisis, sin perder información significativa de los detalles morfológicos de la nube de puntos. Esto se logra eligiendo a la fracción del conjunto de puntos de cada clúster que tengan las ponderaciones mayores, lo cual, además, significa quitar automáticamente a los puntos atípicos.

CAPÍTULO 4. EXPERIMENTACIÓN

4.1 EJEMPLO NUMÉRICO DEL USO DE LOS CUATERNIONES E INTEGRACIÓN EN UN SÓLIDO.

La Tabla IV muestra la relación entre cada una de las caras vistas de un paralelepípedo y su cuaternión unitario correspondiente. La convención del número de cara se indica en la Figura 16.

| Tabla IV. Relación entre el número de cara y su correspondiente cuaternión unitario. | | | |
|---|--------------------------------------|-------------------------------|--|
| CARA | DIRECCIÓN DEL EJE DE ROTACIÓN | ROTACIÓN (En radianes) | CUATERNIÓN UNITARIO CORRESPONDIENTE |
| 1 | (0,1,0) | $-\pi/2$ | $q_1 = (0.7071, 0, -0.7071, 0)$ |
| 2 | (0,0,1) | 0 | $q_2 = (1, 0, 0, 0)$ |
| 3 | (0,1,0) | $\pi/2$ | $q_3 = (0.7071, 0, 0.7071, 0)$ |
| 4 | (1,0,0) | $-\pi/2$ | $q_4 = (0.7071, -0.7071, 0, 0)$ |
| 5 | (1,0,0) | $\pi/2$ | $q_5 = (0.7071, 0.7071, 0, 0)$ |

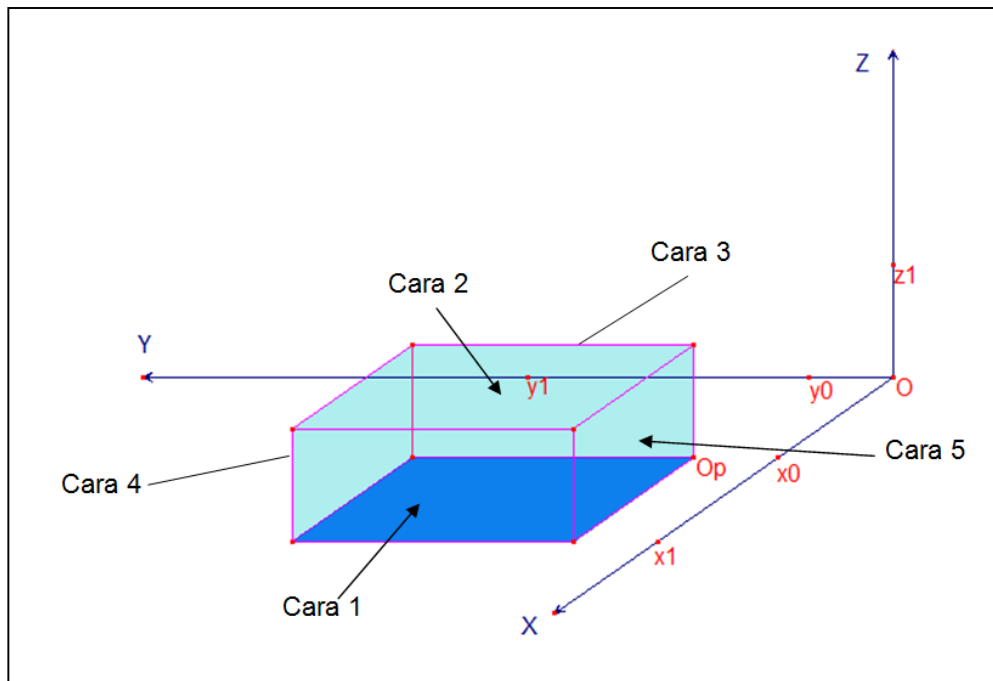


Figura 16. Las 5 caras vistas de un paralelepípedo

La Figura 16 muestra un ejemplo del relieve de las tres caras vistas en escala de grises, con la convención de que mientras más oscura sea la zona, más alta resultará en el relieve. La cara es de base cuadrada de 10 cm de lado. Los cuadros oscuros de las cuatro esquinas poseen una altura de alrededor de 2 cm y el cuadro blanco del centro posee una profundidad de 2 cm con respecto a la base plana (en un tono gris intermedio)

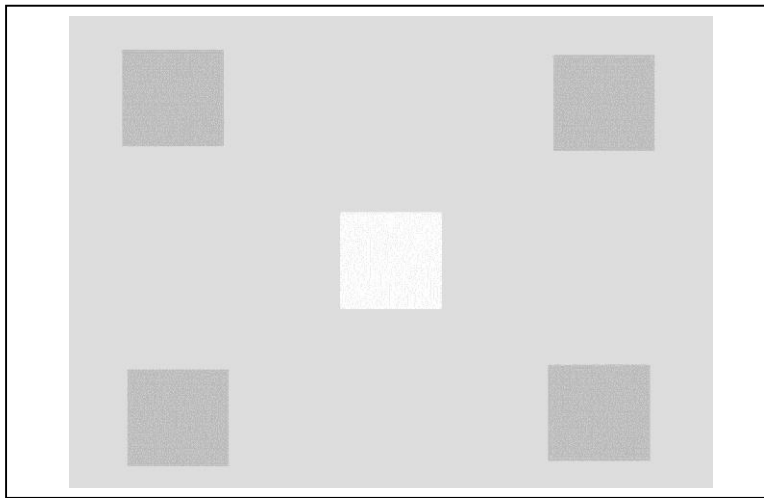


Figura 17. El relieve de las tres caras vistas del paralelepípedo en escala de grises.

La Figura 17 suplirá en este experimento a los datos que se obtengan del escáner láser, por simplicidad, y con la intención de ilustrar el proceso solamente. Sin embargo está claro que la topografía de cada cara puede variar. En el Anexo 1 se muestra el archivo de Matlab con el que se obtiene la Figura 18. El tiempo medio que requiere Matlab 7, con un procesador @ 2 GHz para integrar la figura en 3D utilizando la modelación descrita en el apartado 3.3, es de 7.5 segundos, considerando que las matrices $[X,Y,Z]$ de cada cara son de tamaño de 90x120.

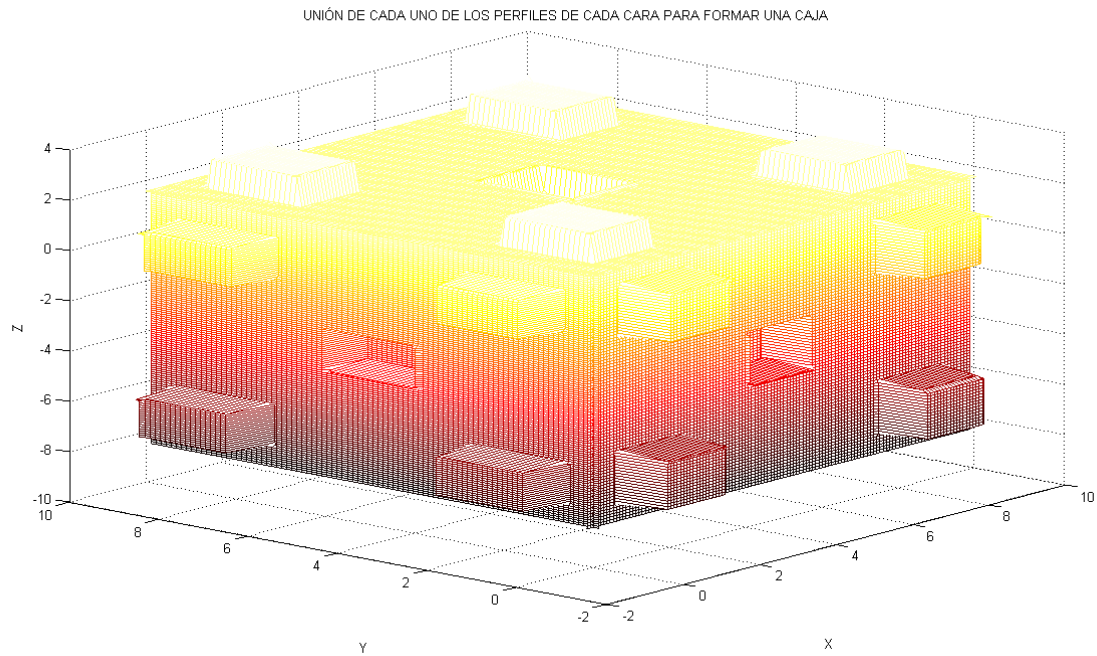


Figura 18. Representación en Matlab de las caras 1,2 y 5, suponiendo que cada una de ellas tiene el aspecto de la Figura 17.

En la Figura 18 se integró un sólido geométrico empleando el algoritmo #1 a partir de información artificial. Sin embargo, en las siguientes figuras, se emplearon datos reales proporcionados por el escáner láser y el brazo robótico.

En la Figura 19 se muestra la gráfica generada con este algoritmo al ser implementado en Matlab.

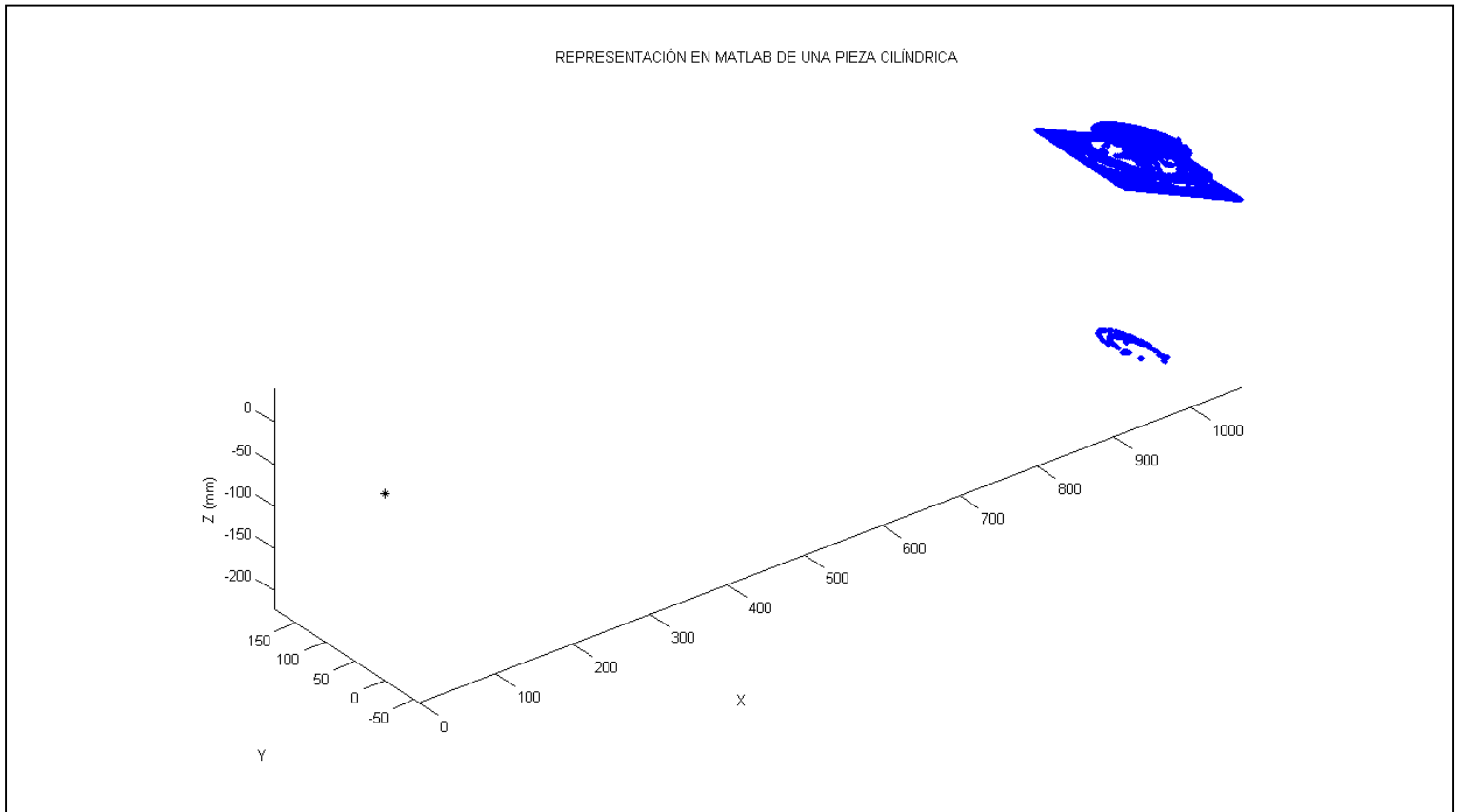


Figura 19. Representación gráfica hecha en Matlab, de una pieza cilíndrica, a partir de la información aportada por el escáner láser y el brazo robot, teniendo al escáner de manera superior.

El punto negro a la izquierda de la Figura 19, indica el origen del sistema de coordenadas global (SCG), el cual, en el presente experimento, se ubicó en la base del robot (valor por default). Los valores de la escala en los ejes X, Y & Z están en

milímetros. Se observa que la representación gráfica del objeto aparece inclinada. Esto es debido a que en el momento en que fue hecho el escaneo, la mesa de inspección no estaba del todo horizontal, y diferencias no perceptibles al ojo humano, de apenas algunos milímetros, en los extremos de la mesa, se hacen evidentes en este momento. Sin embargo, es posible corregir dicha inclinación de “manera lógica”, mediante una transformación lineal. Obsérvese además como en la parte inferior de la representación aparecen algunos puntos atípicos que no existen realmente en la pieza cilíndrica. En el Anexo 3 se muestran algunas rutinas en Matlab para solucionar estos defectos en la presentación de los datos proveídos por el escáner láser.

Cabe señalar que en este experimento, se realizaron 75 disparos, es decir, el valor de m en el algoritmo #1 es 75. El tiempo de ejecución de la rutina del algoritmo #1 implementada en Matlab osciló entre los 10 segundos. En el Anexo 4 se muestra la matriz M que se utilizó en el presente experimento. En la Figura 20 se muestra la fotografía al momento del escaneo de manera superior.

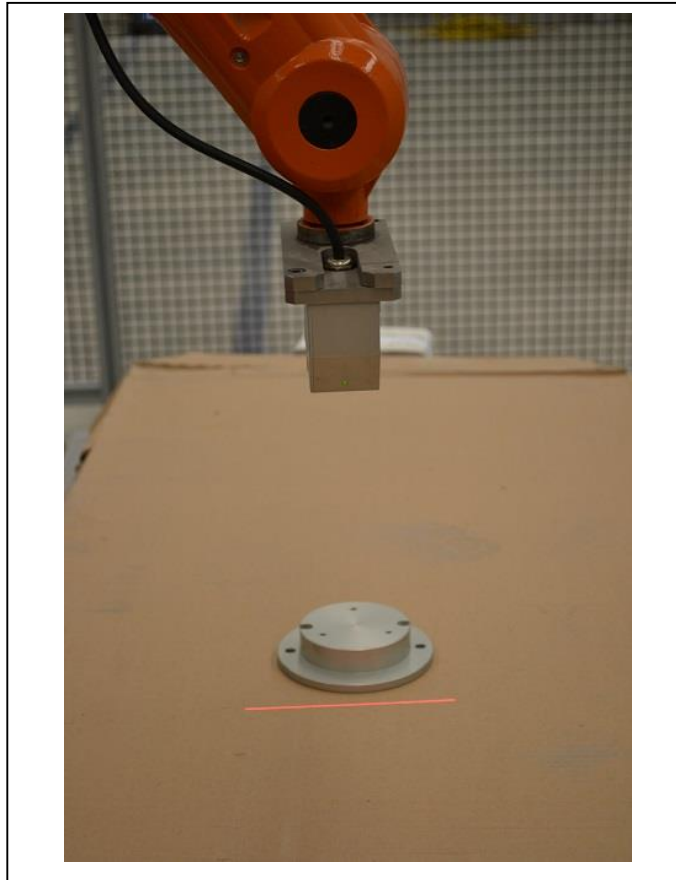


Figura 20. Fotografía del momento del escaneo de la pieza cilíndrica con el escáner láser de manera superior.

En la Figura 21 se muestra otra representación de la misma figura cilíndrica, mostrada en la Figura 19 y 20, sólo que ahora tomando un escaneo de manera lateral (ver Figura 19).

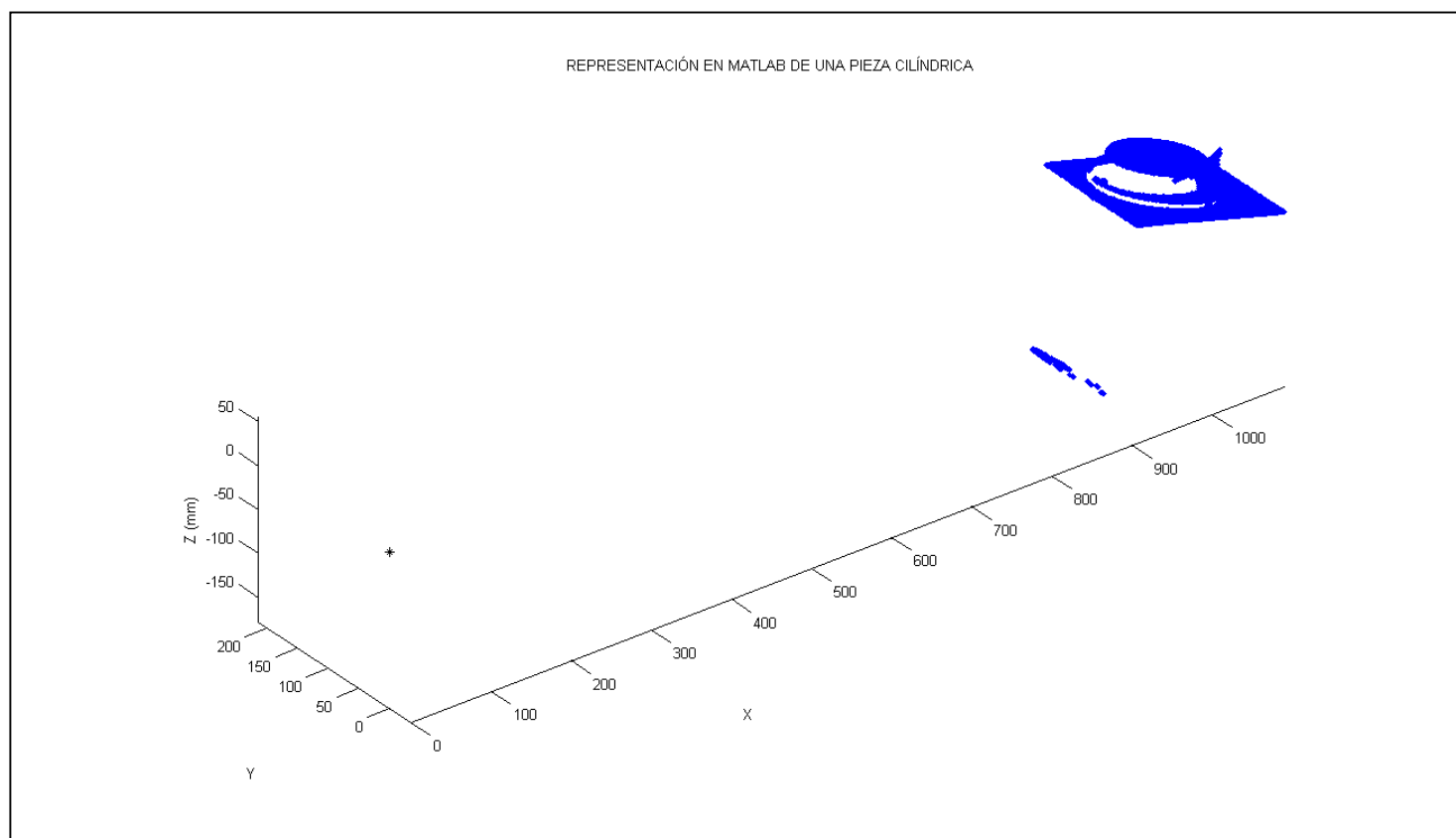


Figura 21. Representación gráfica hecha en Matlab, de una pieza cilíndrica, a partir de la información aportada por el escáner láser y el brazo robot, teniendo al escáner de manera lateral 1.

Se observa que, a diferencia de la Figura 19, en la Figura 21 aparece una región en blanco en la parte izquierda de la representación de la Figura, debido a

que esta parte de la pieza es no visible por el escáner desde el ángulo en que se encuentra este.

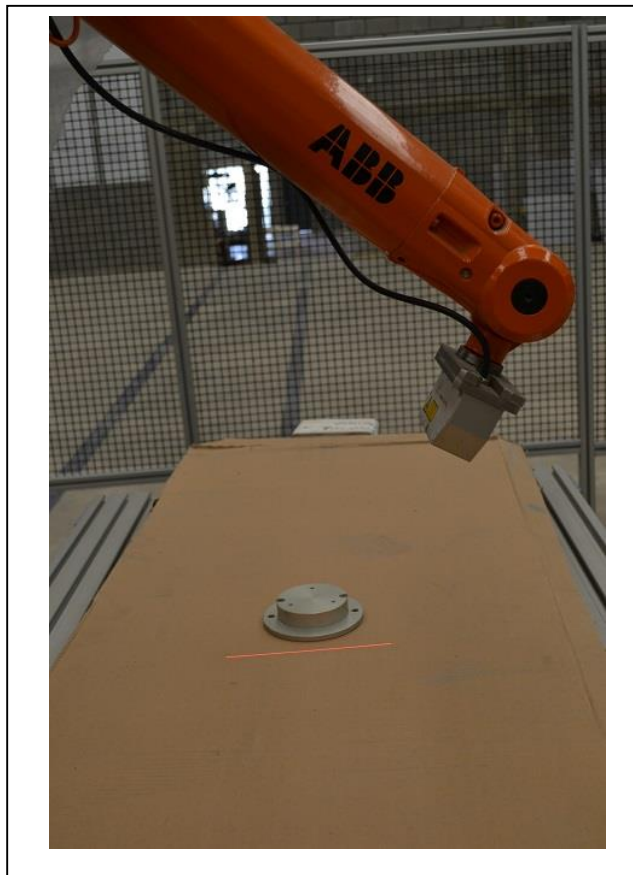


Figura 22. Fotografía del momento del escaneo de la pieza cilíndrica con el escáner láser de manera lateral 1.

En las Figuras 23 y 24 se muestra la contraparte de las Figuras 21 y 22. Es decir, se escaneó la misma pieza cilíndrica de manera lateral opuesta (Lateral 2).

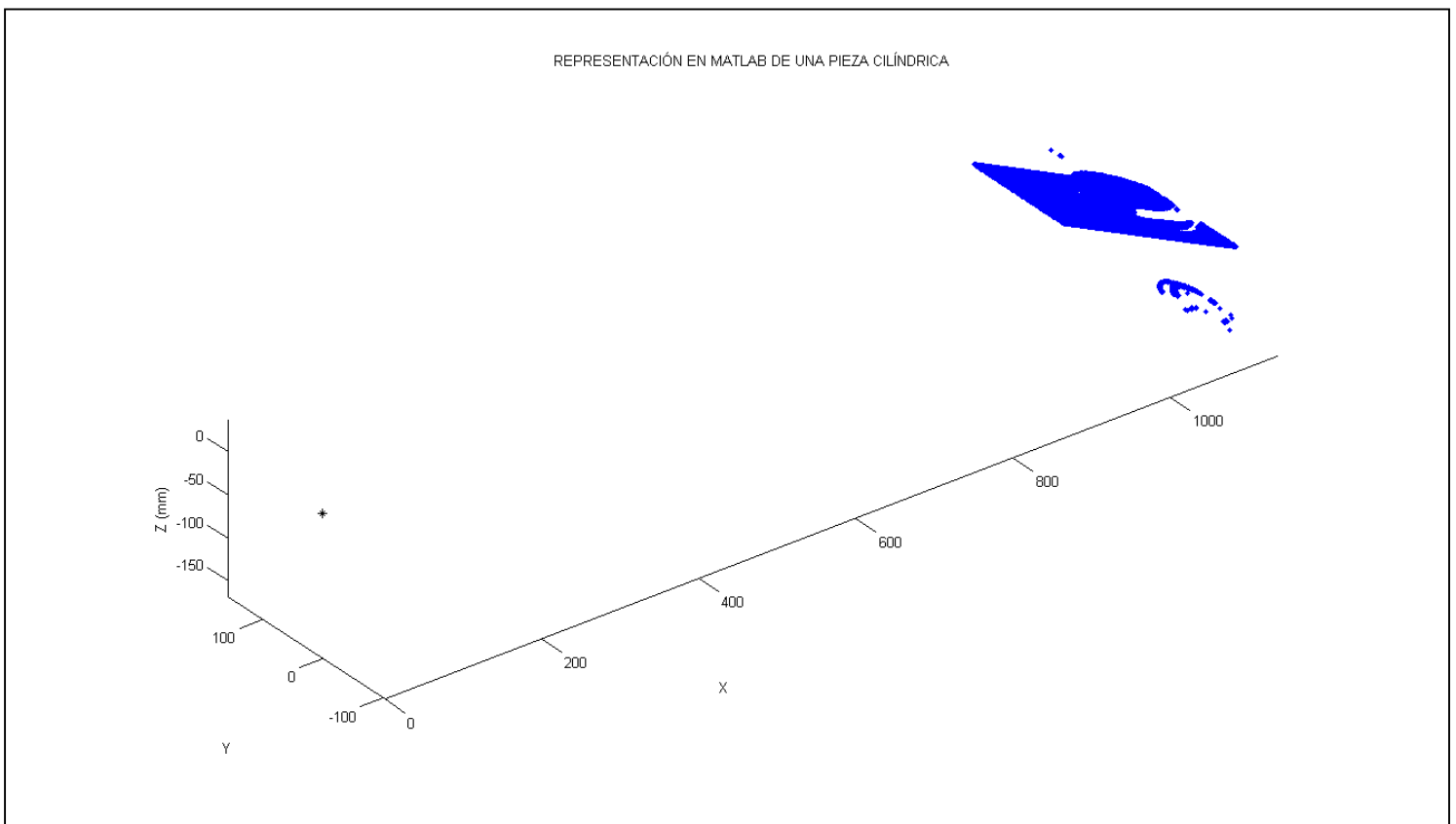


Figura 23. Representación gráfica hecha en Matlab, de una pieza cilíndrica, a partir de la información aportada por el escáner láser y el brazo robot, teniendo al escáner de manera lateral 2.

De manera similar, ahora, aparece una parte en blanco pero en el extremo opuesto al escáner, ya que esta región es no visible para este. Note como en las

tres Figuras, 16, 18 y 20, la representación gráfica de la pieza cilíndrica aparece con la misma inclinación. Eso es debido a que en ningún momento se hubo movido a la pieza cilíndrica o a la mesa durante los tres escaneos.

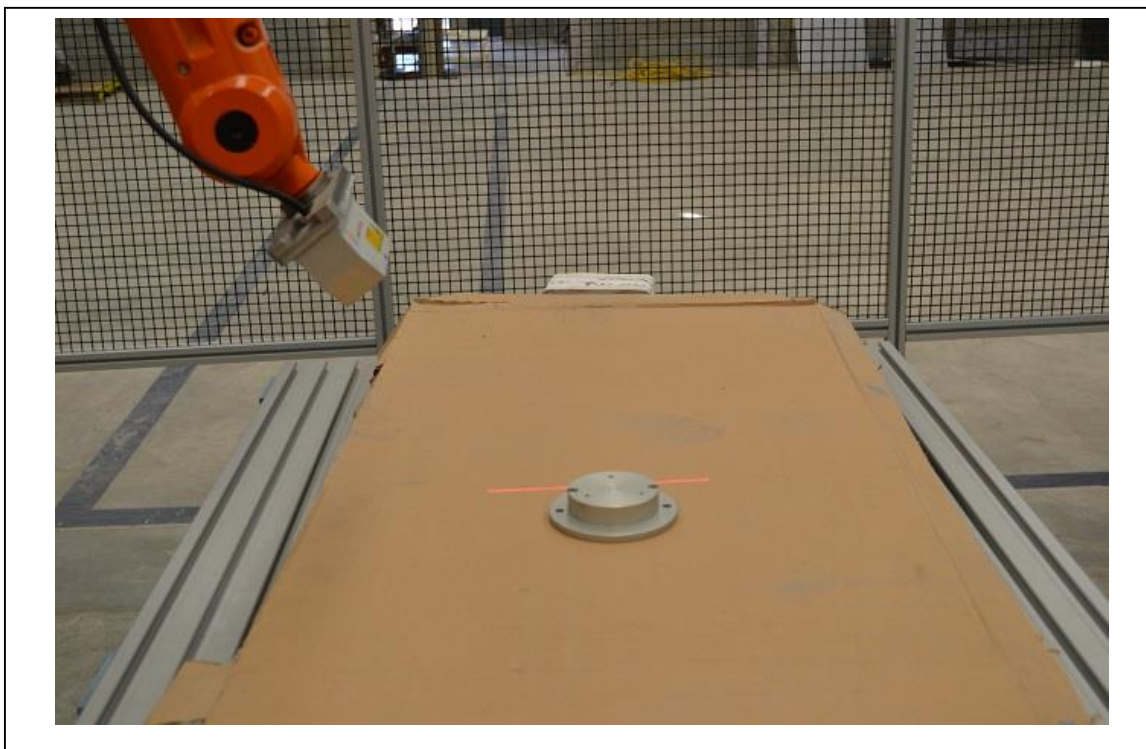


Figura 24. Escaneo de manera lateral 2 de la pieza cilíndrica.

De manera teórica, se podría esperar que superponiendo las tres Figuras: 19, 21 y 23, se podría obtener una representación más fiel y confiable de la pieza escaneada. Sin embargo, como se muestra en la Figura 25, surgen ciertas dificultades técnicas:

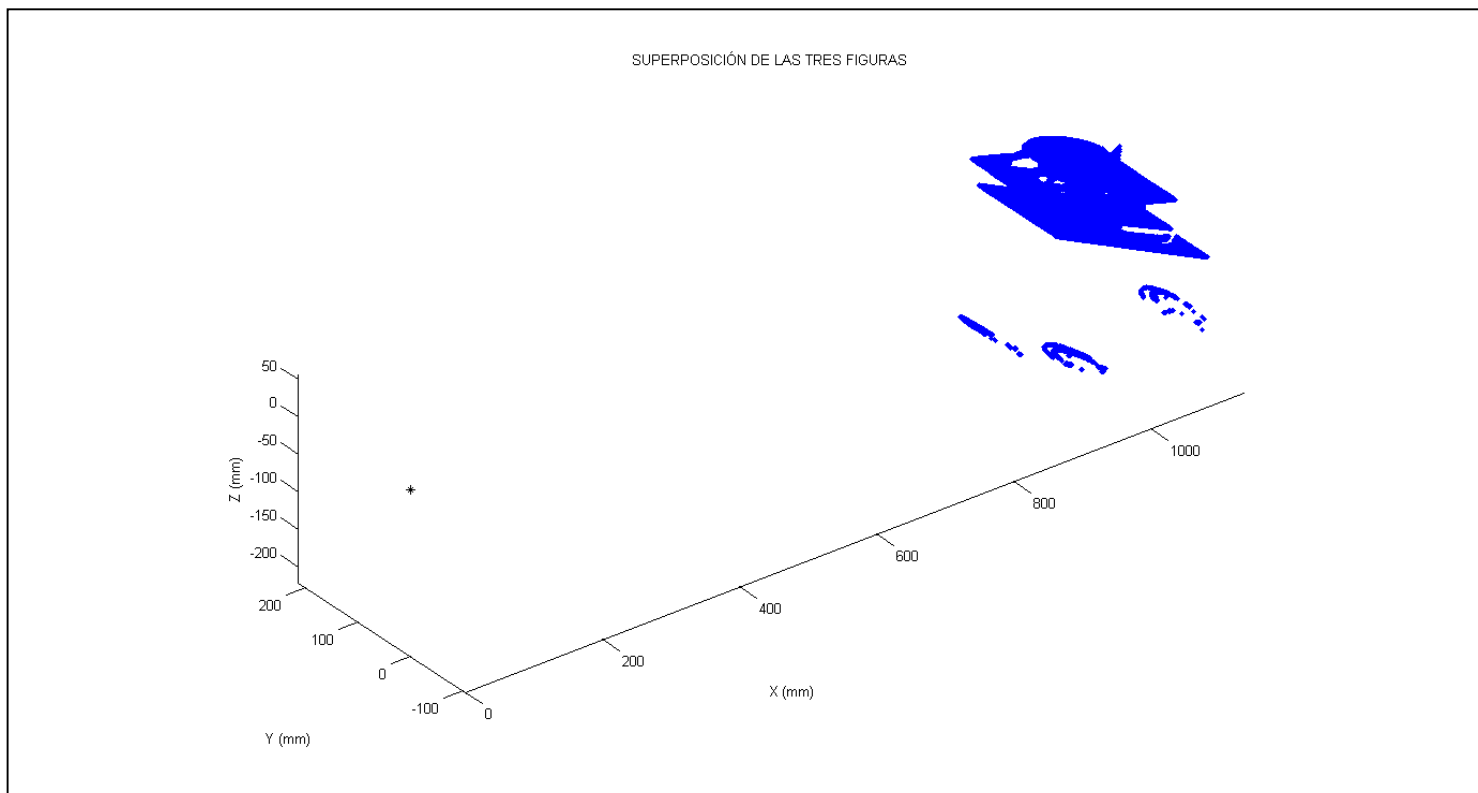


Figura 25. Superposición de las tres Figuras 19, 21 y 23

Como se observa en la Figura 25, las tres Figuras 19, 21 y 23 no se solapan y complementan perfectamente, sino que existe un desfase en la orientación. Esto es debido a las cifras significativas con las que el robot presenta a los valores de los cuaterniones (ver Anexo 4). Al calcular la norma de cada cuaternión, de acuerdo a la ecuación (2), se observa que esta no es *exactamente* igual con 1, si no que difiere

en algunas diezmilésimas. Estas discrepancias originan que al momento de efectuar el acoplamiento de las figuras, estas resulten desacopladas.

4.2 DISCRIMINACIÓN DE PUNTOS ATÍPICOS Y REDUCCIÓN DE UNA NUBE DE PUNTOS.

Con el objeto de disminuir la carga computacional, es preciso reducir el número de la nube de puntos con la consigna de no disminuir los detalles finos del objeto que se escanea. Además, es muy deseable suprimir a los puntos atípicos, que no corresponden a la realidad del objeto y que son producidos entre otras cosas por reflejos ópticos. Se diseñó una rutina en Matlab donde se implementó un algoritmo para:

1. Eliminar puntos atípicos empleando análisis de componentes principales ponderadas (WPCA), particularmente con la ecuación (12), y
2. disminuir el número de puntos de la nube empleando un muestreo uniforme y considerando a los puntos con una mayor ponderación w_i .

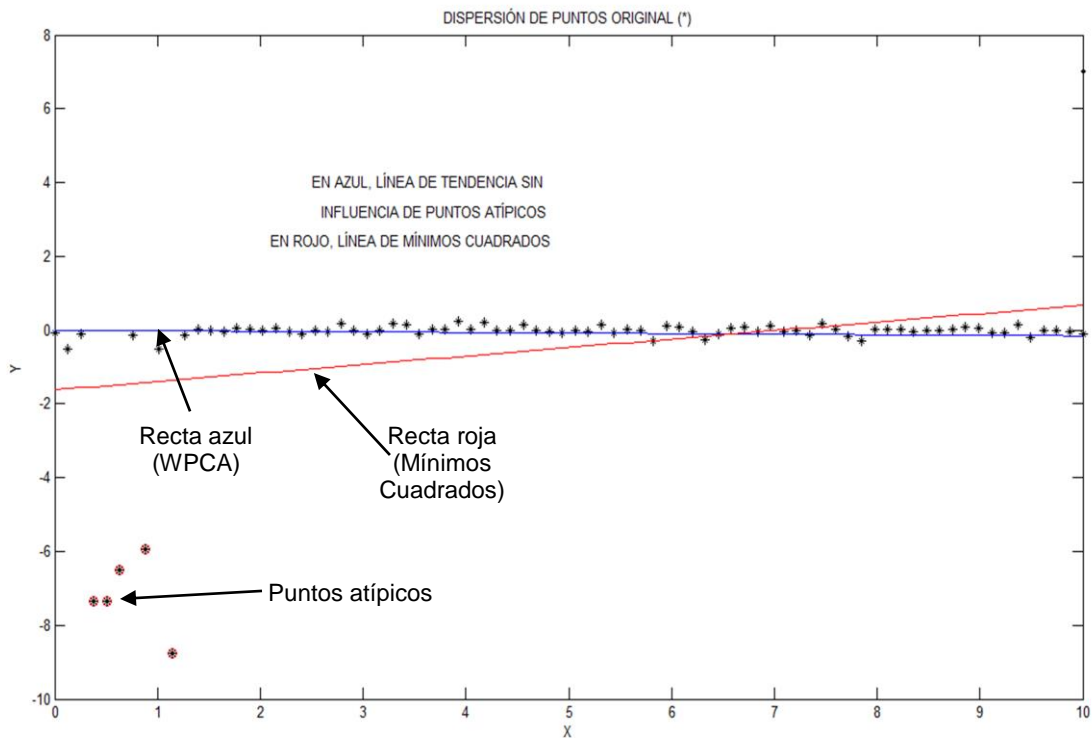


Figura 26. La línea de tendencia azul, obtenida con el algoritmo WPCA y la ecuación 10, ajusta mejor a la tendencia de los puntos sin tomar en cuenta la tendencia de los puntos atípicos (circulados en rojo). En rojo, la recta de regresión como comparación. (80 puntos en total).

El conjunto de puntos mostrado en la Figura 26, se ha construido de tal manera que asemeje a los datos aportados por el escáner, que por lo regular, arroja puntos atípicos en los extremos angulados de perfiles rectos debido a la difracción del haz láser. En la Figura 27 se muestra el resultado de correr la misma rutina pero con datos reales.

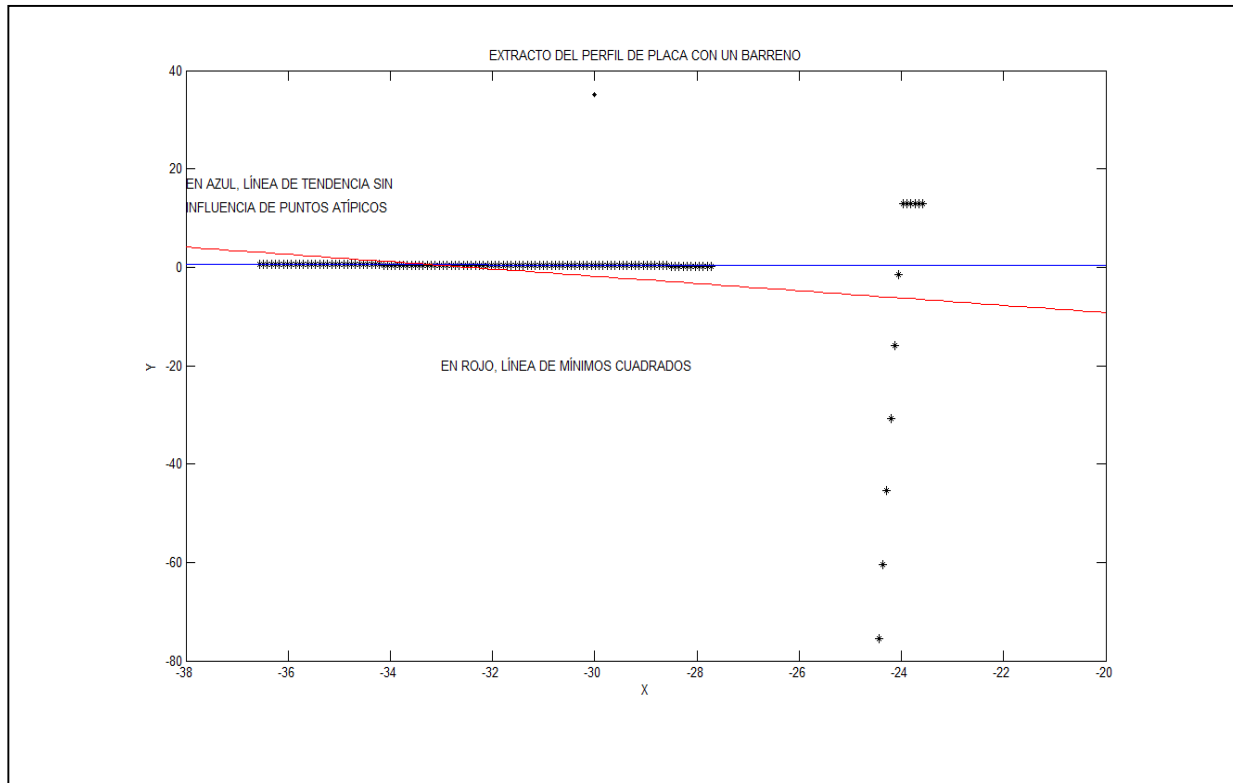


Figura 27. La línea de tendencia azul, obtenida con el algoritmo WPCA y la ecuación 10, ajusta mejor a la tendencia de los puntos sin tomar en cuenta la tendencia de los puntos atípicos a la derecha. En rojo, la recta de regresión como comparación

En la presente investigación se ha utilizado el comando “kmeans” de Matlab para efectuar la clusterización o agrupamiento de los datos en conjuntos homogéneos de datos. La función *kmeans* de Matlab particiona los datos en *k* grupos mutuamente excluyentes, y devuelve el índice de el grupo al cual ha sido asignada cada observación. Esta función minimiza la suma de las distancias entre todos los puntos del conjunto, a un centroide, donde se tiene un centroide distinto para cada grupo.

La métrica con la que se implementa el algoritmo k – means no necesariamente consiste en la euclideana. En la presente investigación, puesto que la mayoría de los perfiles analizados consisten en segmentos rectos, y en la mayoría de los casos, perpendiculares, se utilizó la “métrica de Manhattan” (“city block” en Matlab), ya que a diferencia de la euclideana, esta métrica permite colocar a los centroides sobre los segmentos rectos de los perfiles, lo que permite aislar regiones rectilíneas individuales, necesarias para un mejor análisis. La comparación se observa en las Figuras 27 y 28.

Definición. Métrica (distancia) de Manhattan

Sean $P, Q \in \mathbb{R}^2$, tales que $P = (x_1, y_1)$ y $Q = (x_2, y_2)$. Se define a la distancia de Manhattan, o del taxista, como la función $d: \mathbb{R}^2 \rightarrow \mathbb{R}$, tal que

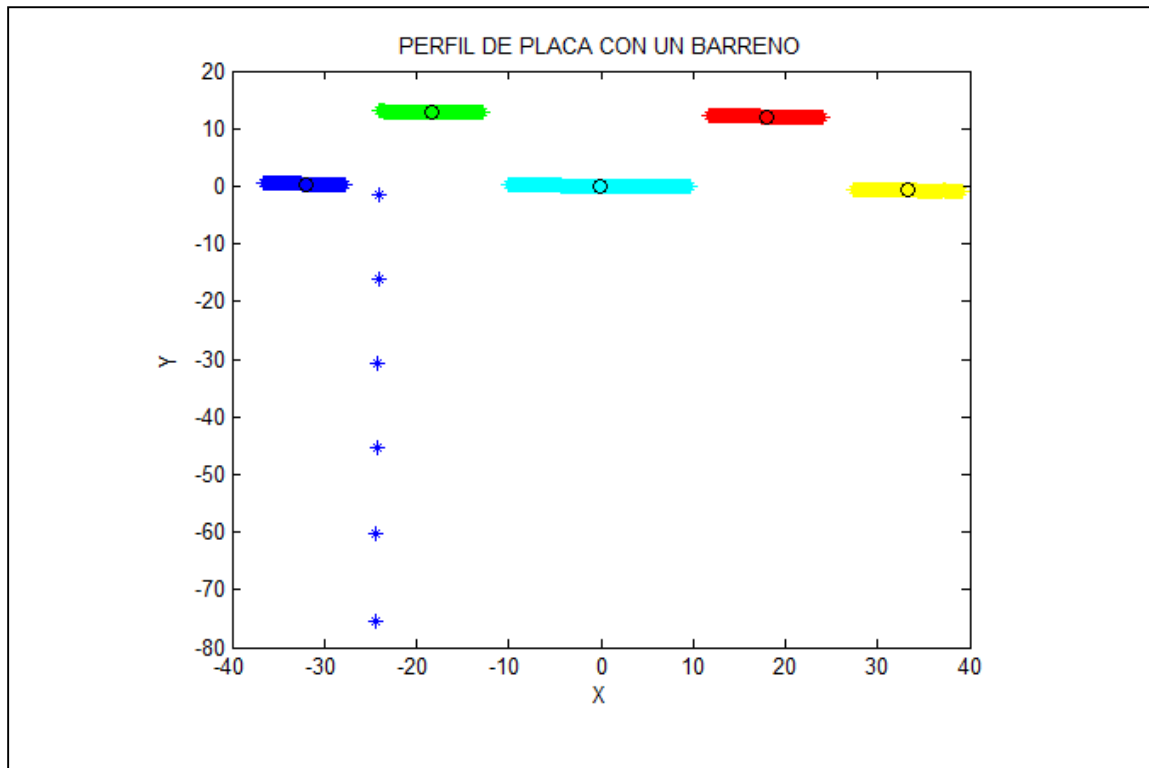
$$d(P, Q) = |x_1 - x_2| + |y_1 - y_2|$$


Figura 28. Determinación de regiones homogéneas mediante el algoritmo k – means con la métrica “cityblock”. (k = 5)

En la Figura 28, el perfil mostrado es el de una pieza plana con un barreno. Las tres regiones de abajo corresponden a la mesa, siendo el espacio entre las dos de arriba el espacio del barreno. La Figura 28 consta de 820 puntos pero observe que no todos son representativos de la pieza metálica.

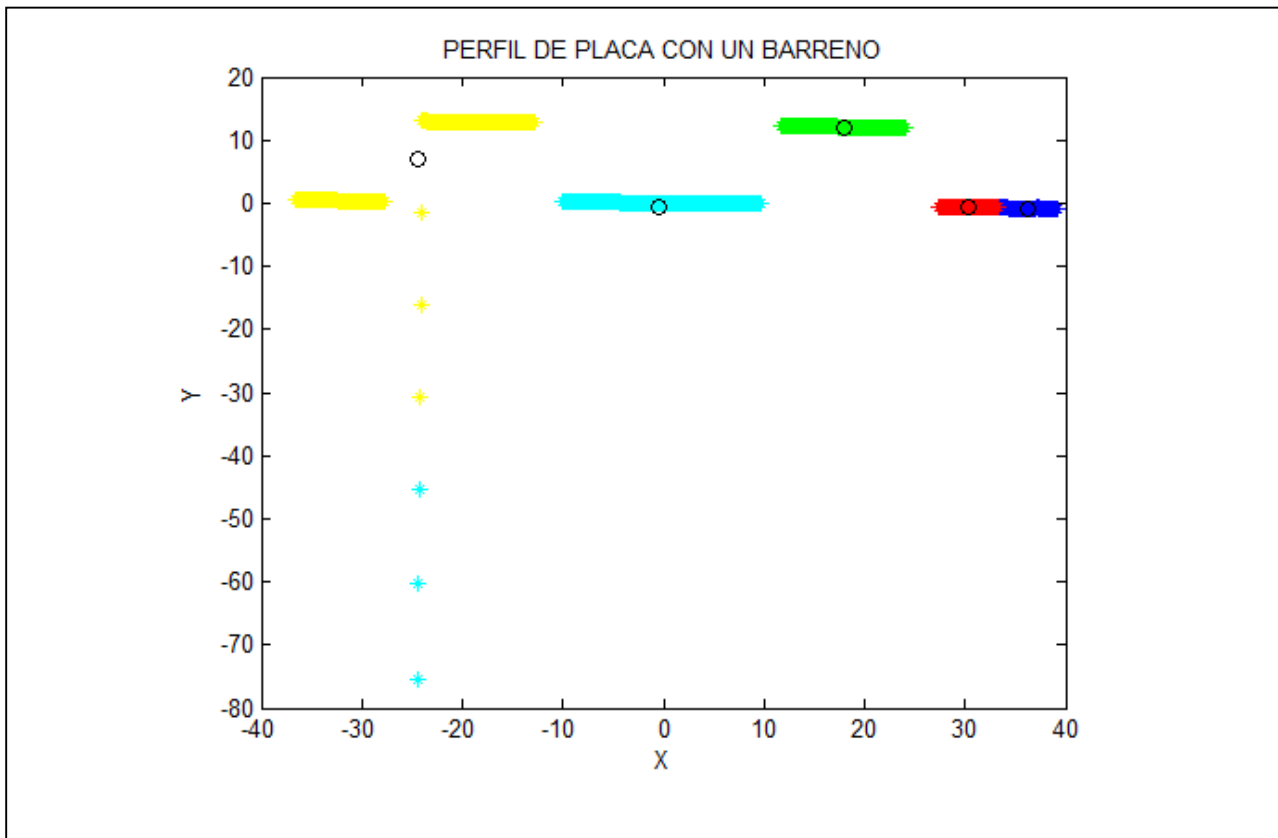


Figura 29. Determinación de regiones homogéneas mediante el algoritmo k – means con la métrica euclídeana. Obsérvese como con la métrica euclídea, el algoritmo k – means considera que las primeras dos líneas de puntos conforman un solo subgrupo con su centroide a la mitad, mientras que en la cuarta línea hay dos subgrupos. ($k = 5$)

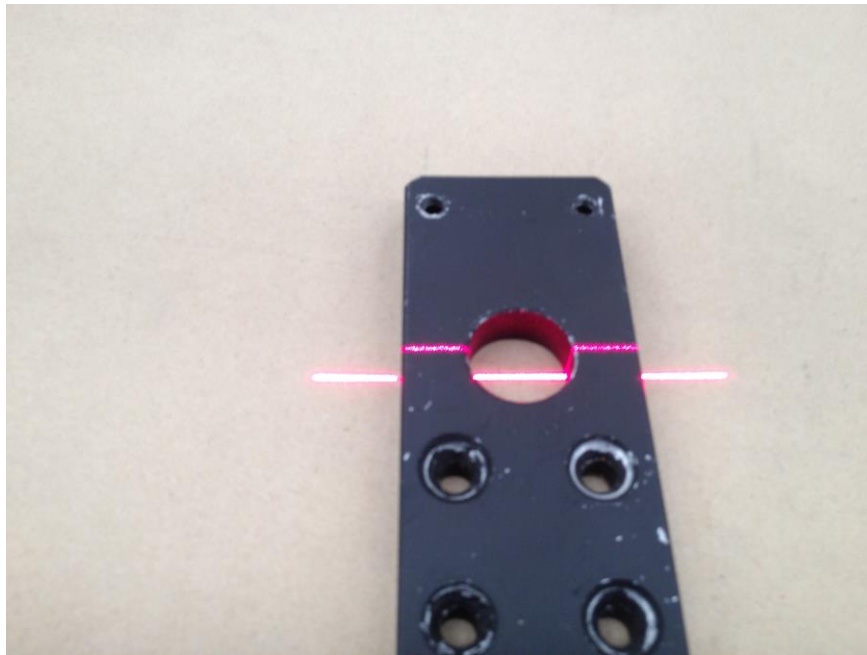


Figura 30. Fotografía del perfil obtenido en la Figura 28

4.3 DIFICULTADES TÉCNICAS EN LA RECONSTRUCCIÓN TRIDIMENSIONAL.

A pesar de la simplicidad del modelo planteado en el Capítulo 3, se encontraron ciertas dificultades técnicas a la hora de implementarlo en una rutina computacional. Las dos dificultades principales encontradas son:

1. **Perfiles aportados por el escáner láser incompletos (Figura 31, Figura 32).** En ocasiones, el número de puntos proveídos por el escáner es menor que el número preestablecido por convención de 1024 puntos.
2. **Presencia de puntos atípicos (Figura 31 y 32).** Además de la falta de puntos relevantes en los perfiles dados por el escáner, se tiene además la presencia de ciertos puntos que se salen de la morfología real del objeto escaneado.

De esta manera, en el Anexo 3, se muestran algunas rutinas diseñadas para satisfacer dichas demandas. Por *estandarización* de los perfiles se entenderá completar el número de puntos por perfil a 1024 (el máximo para nuestro escáner láser) siguiendo la tendencia de los puntos propios del objeto, es decir, tomando en cuenta solamente aquellos puntos que describan el perfil real del objeto.

Es importante mencionar esto, ya que existen ocasiones en los que se registran, además de los puntos de los perfiles de los objetos, los puntos de la mesa

de inspección o de los mecanismos de sujeción de las piezas. El proceso de ubicar puntos atípicos hace referencia a la localización e identificación de puntos que aunque aparezcan en el perfil, no reflejan la realidad del perfil del objeto inspeccionado. Una vez que se han ubicado dichos puntos, deben de discriminarse del posterior procesamiento de los datos, para que este sea más confiable.

En la Figura 31 se diseñó un algoritmo para completar los perfiles a 1024 puntos, siguiendo la tendencia de los puntos de acuerdo a la morfología de la superficie de este, a fin de estandarizarlos y estar en posibilidades de continuar con la reconstrucción tridimensional. En tal ejemplo sólo se enfocó al caso de completar con líneas rectas (en rojo).

En la Figura 32 se muestra el resultado de un algoritmo que identifica y discrimina a los puntos atípicos de los perfiles suministrados por el escáner láser y se completó un perfil circular tomando en cuenta solamente a algunos puntos con tendencia circular en el perfil.

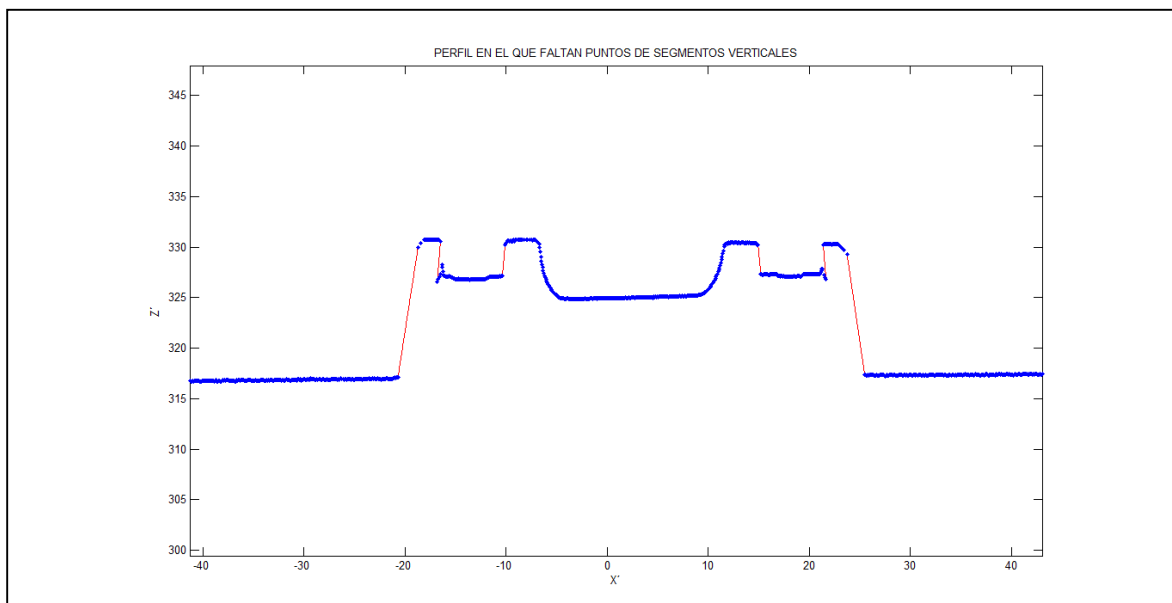


Figura 31. Perfil obtenido por el escáner (puntos azules) en el que faltan puntos de los segmentos verticales. De los 1024 puntos teóricos, sólo muestra 996. En rojo, los segmentos verticales faltantes obtenidos con el algoritmo #3 de los anexos.

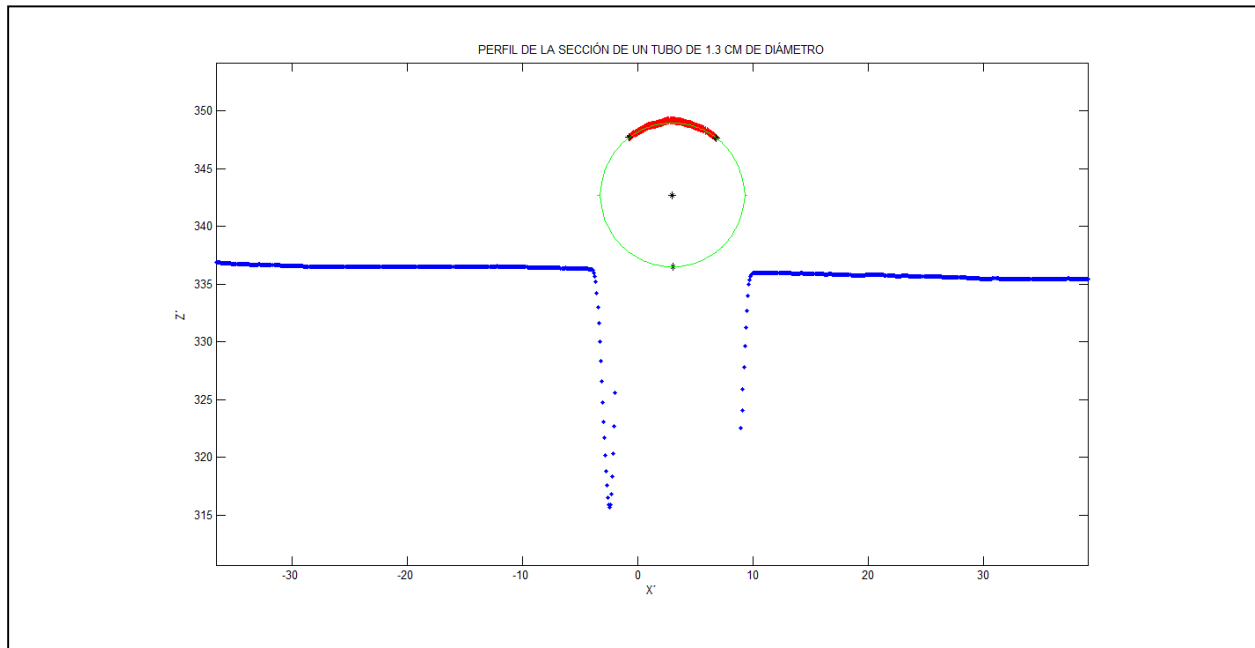


Figura 32. Perfil obtenido por el escáner (puntos azules) de una varilla cromada de 1.3 cm de diámetro con un tiempo de exposición de 10 segundos. Observe cómo los segmentos horizontales corresponden a la mesa de inspección y sólo la parte roja remarcada corresponde a la pieza. De los 1024 puntos teóricos, sólo muestra 926. Debajo de la circunferencia construida se observan además algunos puntos atípicos.

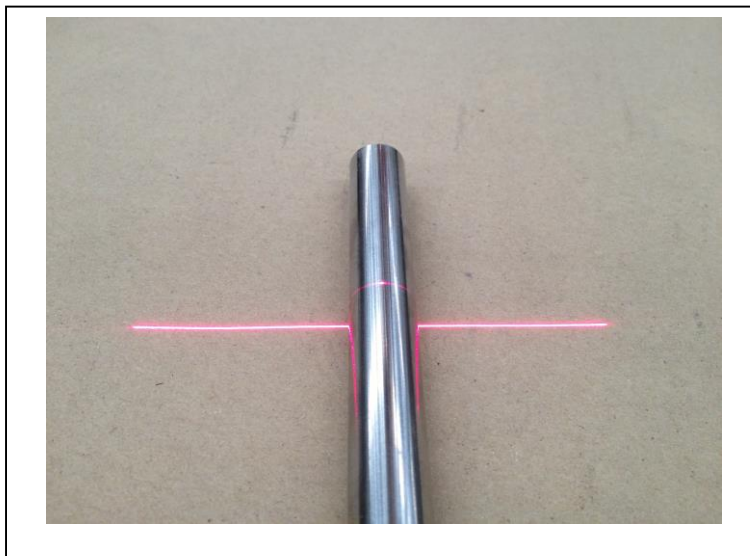


Figura 33. Fotografía de la pieza escaneada en la Figura 32.

CAPÍTULO 5. CONCLUSIONES

5.1 CONCLUSIONES

Se ha observado que la modelación matemática desarrollada en el Capítulo 3, ha dado resultados a la hora de ser implementada en Matlab. Este trabajo representa un primer paso en el diseño de un sistema flexible, que pueda ser adaptado en línea para satisfacer las necesidades de inspección de las diferentes industrias manufactureras. Se logró obtener un modelo tridimensional de una pieza de forma cilíndrica, a partir de de la integración de perfiles bidimensionales.

El tiempo de ejecución fue razonable, ya que para la integración de cualquiera de las Figuras 21, 23 o 25, el tiempo máquina no superó los 11 segundos, siendo, además, implementado el sistema en un equipo portátil bastante modesto.

Se pretende realizar la comparación con un modelo estructurado previamente para determinar si la pieza cumple con criterios de calidad, para que a partir de dicha información, la pieza sea aceptada o rechazada.

5.2 RECOMENDACIONES Y TRABAJO FUTURO.

Como trabajo futuro se planea diseñar un algoritmo que se pueda implementar para inspeccionar piezas grandes, tales que requieran más de un barrido por cara y la inclusión de una banda transportadora que permita utilizar el sistema en planta de manera práctica. Así mismo, queda pendiente el diseño y la implementación del algoritmo que permita comparar el modelo 3D así obtenido con un modelo estandarizado y predeterminado.

Para lograr una superposición perfecta de las imágenes de una misma figura obtenidas a distintas orientaciones, se recomienda emplear matrices de transformación homogénea en lugar de cuaterniones.

Así mismo, otra recomendación, con respecto a seleccionar algunos puntos de cada clúster, para reducir el número de la nube de puntos, pero manteniendo la misma tendencia del grupo, consiste en efectuarlo mediante un algoritmo genético.

ANEXO #1: Rutina en Matlab para cambiar la orientación espacial empleando cuaterniones. (Figura 18)

```

clc, clear, clf

A=imread('C:\Users\HP\Desktop\Documentos Demaq\imagenes\p3.jpg');
%captura la imagen matriz de (mxn)x3   R G B
A1=double(A); %convierte a escala de grises y cambia a formato matriz mxn
double, en lugar de uint8, si la imagen es a color, usar rgb2gray antes
Z=A1(1:8:end,1:8:end); % reduce el tamaño de la matriz A1 sin perder
información; Z es submuestreo de A1
[m n]=size(Z); %tamaño de la matriz Z mxn
[X Y]=meshgrid(linspace(0,10,n),linspace(0,10,m));% crea un rectángulo de
25x25 en el plano X-Y,
zmax=abs(max(max(Z)))% para normalizar la figura
Zinv=10.*((255-Z)./zmax); %invierte la escala de grises, para que los
colores fuertes sean proyectados hacia "arriba" en lugar de hacia "abajo"
axis equal

tic
%+++++++CARA #1 ++++++
theta=-pi/2;%input('introduce el ángulo de rotación theta = ');
x=0;%input('introduce la componente de la dirección x = ');
y=1;%input('introduce la componente de la dirección y = ');
z=0;%input('introduce la componente de la dirección z = ');

q=[cos(theta/2) x*sin(theta/2) y*sin(theta/2) z*sin(theta/2)]
q=q/norm(q)
qc=quatconj(q)

for i=1:1:m
    for j=1:1:n
        xx=quatmultiply(quatmultiply(q,[0 X(i,j) Y(i,j) Zinv(i,j)]),qc);
X1(i,j)=xx(1,2);
        yy=quatmultiply(quatmultiply(q,[0 X(i,j) Y(i,j) Zinv(i,j)]),qc);
Y1(i,j)=yy(1,3);
        zz=quatmultiply(quatmultiply(q,[0 X(i,j) Y(i,j) Zinv(i,j)]),qc);
Z1(i,j)=zz(1,4);
    end
end

%+++++++CARA #2+++++++
theta= pi/2;%input('introduce el ángulo de rotación theta = ');
x=1;%input('introduce la componente de la dirección x = ');
y=0;%input('introduce la componente de la dirección y = ');
z=0;%input('introduce la componente de la dirección z = ');

```

```

q=[cos(theta/2) x*sin(theta/2) y*sin(theta/2) z*sin(theta/2)]
q=q/norm(q)
qc=quatconj(q)

for i=1:1:m
    for j=1:1:n
        xx=quatmultiply(quatmultiply(q,[0 X(i,j) Y(i,j) Zinv(i,j)]),qc);
X2(i,j)=xx(1,2);
        yy=quatmultiply(quatmultiply(q,[0 X(i,j) Y(i,j) Zinv(i,j)]),qc);
Y2(i,j)=yy(1,3);
        zz=quatmultiply(quatmultiply(q,[0 X(i,j) Y(i,j) Zinv(i,j)]),qc);
Z2(i,j)=zz(1,4);
    end
end

toc

mesh(X,Y,Zinv)
colormap('hot')
hold on
Z1=Z1-8.5;% operaciones de ajuste para que las aristas de cada cara
coincidan
X1=X1+1.7; %ajuste
mesh(X1,Y1,Z1)
hold on
Z2=Z2-8.5;
Y2=Y2+1.7;
mesh(X2,Y2,Z2)
title('A SOLID BOX CONSTRUCTED BY THE SAME SURFACE REORIENTED ACCORDING
TO ITS QUATERNION')
xlabel('X')
ylabel('Y')
zlabel('Z')

```


ANEXO #2: Rutina en Matlab para la reconstrucción 3D a partir de perfiles bidimensionales. (Figuras 19, 21, 23 y 25)

```

clc, clear, clf
tic
M=xlsread('C:\Users\HP\Desktop\Documentos Demaq\Figura
Cilindrica.xlsx','Superior','a3:caa403');
[m n]=size(M);
h=500;
for j=1:m
Xscan=M(j,1);
Yscan=M(j,2);%Coordenadas del escáner con respecto al robot
Zscan=M(j,3);
q1=M(j,7);
q2=M(j,6);% Valores de los cuaterniones actuales
q3=M(j,5);
q4=M(j,4);

Pprim=[M(j:j,8:1031)' (h - M(j:j,1032:2055))']; %Coordenadas de los
puntos dados por el escáner (En x'y z')
Cscan=ones(1024,1)*[Xscan Yscan Zscan]; %Para crear una matriz de
1024x3 donde cada fila es Xscan Yscan Zscan y poder realizar
% operaciones en la fila 37
% (con matrices del mismo
% tamaño)

q=[q1 q2 q3 q4]; %cuaternión
qc=quatconj(q); %cuaternión conjugado
pq1x=quatmultiply(q,[0 1 0 0]);%cálculos para determinar ux y uy
pq2x=quatmultiply(pq1x,qc);

pq1z=quatmultiply(q,[0 0 0 1]);
pq2z=quatmultiply(pq1z,qc);

ux=[pq2x(2) pq2x(3) pq2x(4)]; %vectores unitarios en las direcciones X' y
Z' (Del sistema de coordenadas del escáner)
uz=[pq2z(2) pq2z(3) pq2z(4)];

for i=1:1024
    Xp(i,:)=Pprim(i,1).*ux; %Se multiplican las coordenadas de cada punto
    dado por el escáner por los vectores unitarios ux y uz
    Zp(i,:)=Pprim(i,2).*uz; % para convertir sus coordenadas al sistema
    de coordenadas global
end

P=Xp+Zp+Cscan; %coordenadas de los puntos del escáner transformadas a
las coordenadas del sistema global de coordenadas

```

```

plot3(P(:,1),P(:,2),P(:,3),'.b',0,0,0,'k*'), hold on
%,Cescan(:,1),Cescan(:,2),Cescan(:,3),'b*'

end

axis equal
xlabel('X (mm)'), ylabel('Y (mm)'), zlabel('Z (mm)'),
title('SUPERPOSICIÓN DE LAS TRES FIGURAS')

toc

```

ANEXO #3: Rutina en Matlab para la obtención de una recta que mejor ajuste a un conjunto de puntos con tendencia lineal discriminando puntos atípicos. (Figura 26)

```
clc, clear, clf
x=linspace(0,10,160); %Generación de un segmento recto con puntos
atípicos normalmente distribuidos en un extremo
yy=0.*x;
n=numel(x);
corte=10
for i=1:corte
    eps=-9*rand(1)-0;

    if abs(eps)>5.2
        y(i)=yy(i)+eps;
        plot(x(i),y(i), 'or'), hold on
    else
        y(i)=yy(i)+eps/9;
    end
end
for i=corte+1:n
    eps=randn(1);
    y(i)=yy(i)+eps/10;
end

plot(x,y, '*k'), hold on
%axis equal

%INICIO ALGORITMO WPCA

m=mean(y); %valor medio del conjunto de puntos y
d=abs(y-m);
inv=1./d;
suminv=sum(inv);

for i=1:n
    wi(i)=1/(d(i).*suminv);
end

for i=1:n
    wx(i)=wi(i).*x(i);
    wy(i)=wi(i).*y(i);
end
P=polyfit(wx,wy,1)
Q=polyfit(x,y,1)

plot(x,P(1).*x + P(2), 'b',x,Q(1).*x + Q(2), 'r')
xlabel('X'), ylabel('Y'), title('DISPERSIÓN DE PUNTOS ORIGINAL (*)')
plot(10,7, '.k')
text(2.5,4, 'EN AZUL, LÍNEA DE TENDENCIA SIN')
```

```
text(2.6,3.2,'INFLUENCIA DE PUNTOS ATÍPICOS')
text(2.1,2.4,'EN ROJO, LÍNEA DE MÍNIMOS CUADRADOS')
```

ANEXO #4: Rutina en Matlab para la agrupación de puntos en regiones homogéneas con el algoritmo k – means. (Figura 27)

```
clc, clear, clf
M=xlsread('C:\Users\HP\Documents\MATLAB\Demag mayo\perfiles\plano con
barreno\profile_0008.slk');
m=length(M);
j=1;
% Se extrae una porción de un perfil real a continuación,
% para facilitar la experimentación.
tic
for i=1:m
    if M(i,2)>324 %Altura suficiente para despreciar a la mesa
        Mx(j)=M(i,1);
        My(j)=M(i,2);
        j=j+1;
    end
end

X=Mx(1:1:end); %Muestreos uniformes
Y=My(1:1:end);
n=length(X)
color=1;
%ALGORITMO K - MEANS: Agrupa a el conjunto de puntos en k regiones
%homogéneas
k=5; %número de regiones homogéneas
[w ,C]=kmeans([X' Y'],k,'distance','cityblock');

for h=1:k
    switch color
        case 1
            c1=1; c2=0; c3=0;
            color=color+1;
        case 2
            c1=0; c2=1; c3=0;
            color=color+1;
        case 3
            c1=0; c2=0; c3=1;
            color=color+1;

        otherwise
            color=1;
    end

    for j=1:n
        if w(j)==h
            plot(X(j),Y(j),'.','Color',[c1 c2 c3]), hold on
```

```
        end
    end

    end
plot(C(:,1),C(:,2),'*k')
toc
title('DETERMINACIÓN DE REGIONES HOMOGÉNEAS MEDIANTE EL ALGORITMO K -
MEANS')
xlabel('X'), ylabel('Z')
```

ANEXO #5: Rutinas diversas para la estandarización de perfiles bidimensionales.

1. Enderezado (nivelación) de perfiles.

Se elaboró un código en lenguaje M que es capaz de enderezar los perfiles de manera lógica sin perder la información representativa de los perfiles. Este detalle técnico se ha previsto también ya que en la práctica es muy difícil tener una superficie de inspección, como una mesa, perfectamente horizontal, por lo que si se grafican directamente la información del escáner, los perfiles tienen un aspecto inclinado.

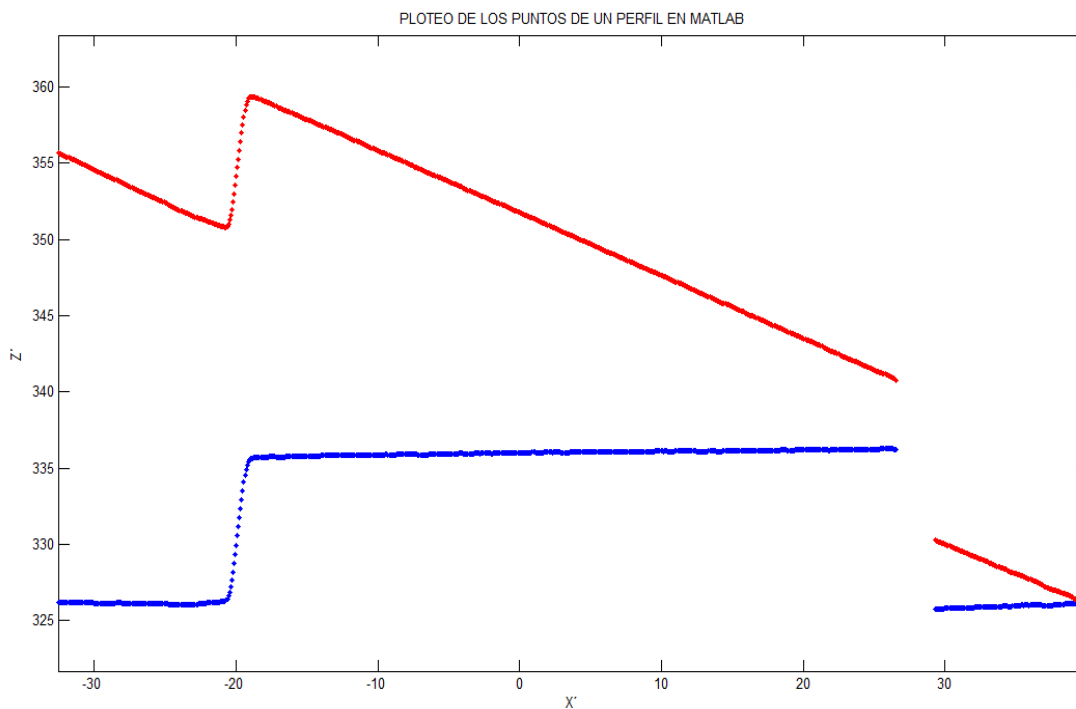


Figura 31. En rojo, el conjunto de puntos tal y como lo da el escáner. En azul, el mismo perfil enderezado

Código en Matlab

```
clear, clf, clc
M=xlsread('C:\Users\HP\Documents\MATLAB\Prueba1\Perfil_0001.xls');
n=size(M)
a=M(1,2); %Se toman los puntos extremos de la gráfica para luego
enderezar la gráfica (ya que por lo general la mesa está inclinada)
b=M(n(1),2);
d=abs(a-b); % diferencia en la altura de los puntos extremos de la
gráfica
%f=1024-n(1)% faltante de puntos para 1024
%m=min(M(:,2))

%las líneas siguientes enderezan la gráfica
if a - b < 0 % gráfica inclinada en forma decreciente
    for k=1:n(1)
        yc(k)=M(k,2) - d*(k-1)/n(1);%así estandariza las unidades del eje
X y Z, las cuales estaran en milímetros
    end
else
    for k =1:n(1) %gráfica inclinada en forma creciente
        yc(k)=M(k,2) + d*(k-1)/n(1);
    end
end

plot(M(:,1),yc,'b.',M(:,1),M(:,2),'r.'), hold on %Gráfica enderezada en
azul
axis equal
xlabel('X'), ylabel('Z'), title('PLOTEO DE LOS PUNTOS DE UN PERFIL EN
MATLAB')
```

2. Traslación vertical u horizontal de perfiles.

Esta aplicación traslada verticalmente y/u horizontalmente a un perfil de puntos, según se especifique por el usuario.

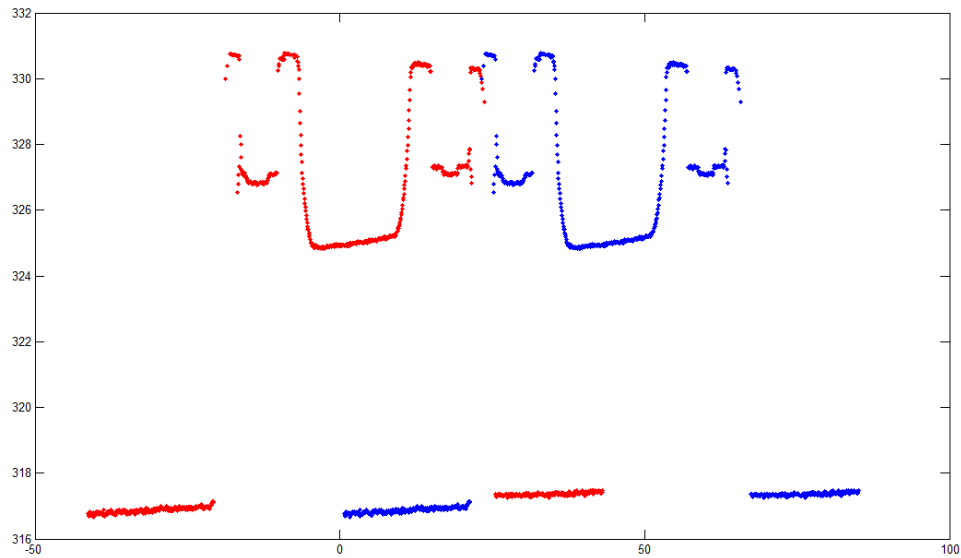


Figura 32. En rojo el plot de los puntos originales, en azul, la gráfica trasladada a cero en x.

Código en Matlab

```
clc, clear, clf
M=xlsread('C:\Users\HP\Documents\MATLAB\Prueba1\Perfil_0001.xls');
h=input('Indica el número de unidades a trasladar horizontalmente (positivo a la derecha, negativo a la izquierda): ');
k=input('Indica el número de unidades a trasladar verticalmente (positivo arriba, negativo abajo): ');
plot(M(:,1),M(:,2),'.r', M(:,1)+h,M(:,2)+k,'.b')
```


3. Completar perfiles con segmentos rectos.

Esta aplicación completa un perfil con huecos, agregando segmentos rectos entre puntos relativamente alejados (distancia mayor que la unidad).

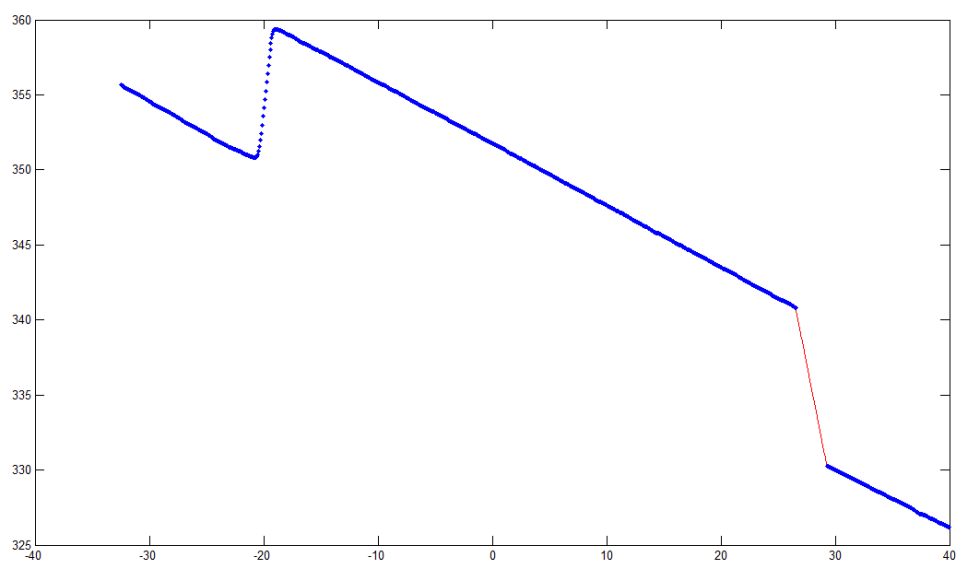


Figura 33. En azul los puntos originales, en rojo, el segmento de recta agregado.

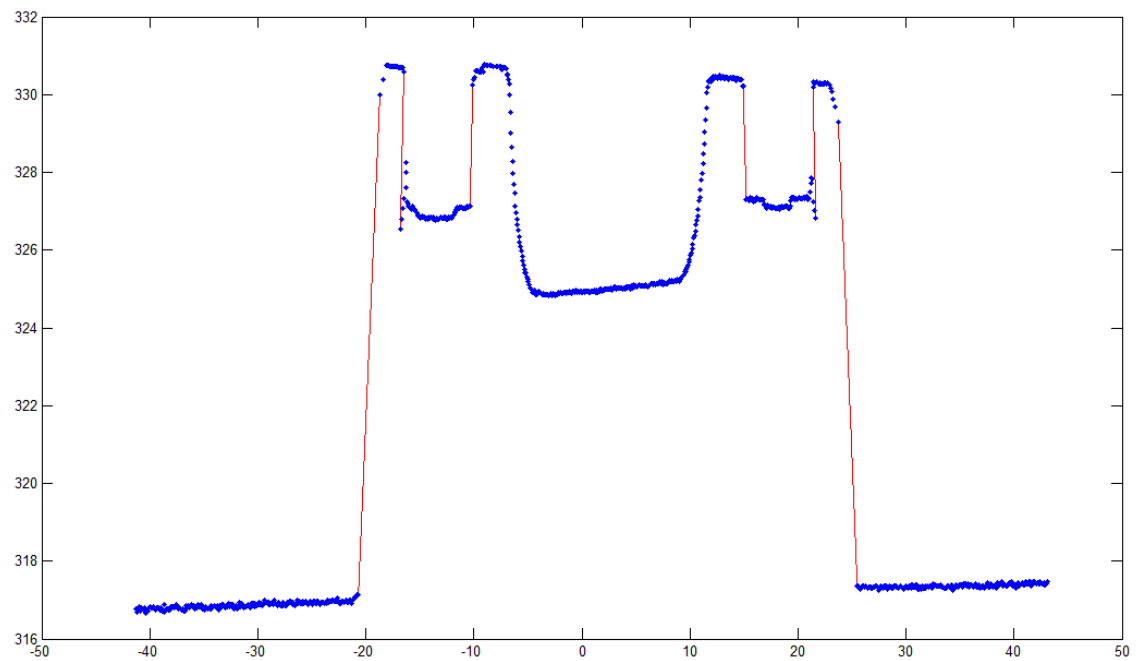


Figura 34. En azul, los puntos originales del perfil, tal como se dan por el escáner, en rojo, los segmentos de recta agregados.

Código en Matlab

```
clc, clear, clf
M=xlsread('C:\Users\HP\Documents\MATLAB\Prueba1\Perfil_0001.xls');
n=size(M)

for i=1:n(1)-1 %Con este "for" se ubican los huecos en la gráfica
(indicando los índices)
    Dx(i)=abs(M(i+1,2)-M(i,2)); %Se obtiene un vector Dx en el que se
registran todas las alturas de puntos adyacentes
end
hx=mode(Dx) % la moda del vector Dx
j=1; I=1;
for i=1:n(1)-1
    if Dx(i)>1 %Se registran los índices para diferencias entre alturas
mayores que la unidad
        I(j)=i;
        j=j+1;
    end
end
I
k=size(I)
for j=1:k(2)
    m=(M(I(j),2)-M(I(j)+1,2))/(M(I(j),1)-M(I(j)+1,1))
    t=linspace(M(I(j),1),M(I(j)+1,1));
```

```
    y=m.*t + M(I(j)+1,2) - m.*M(I(j)+1,1);  
    plot(t,y,'-r'), hold on %En rojo las rectas que se han agregado  
end  
plot(M(:,1),M(:,2),'.'),hold on %En azul, los datos originales
```

4. Cortar un perfil de puntos a partir de una altura indicada.

Esta aplicación recorta una gráfica a partir de una cierta altura especificada por el usuario.

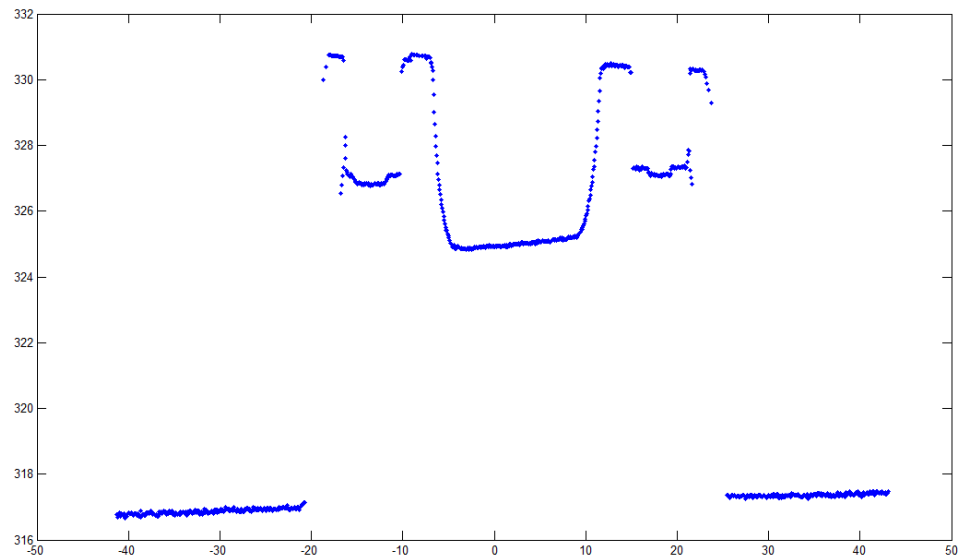


Figura 35. Perfil con datos originales. Se observa que los puntos por debajo de 318 corresponden a la mesa.

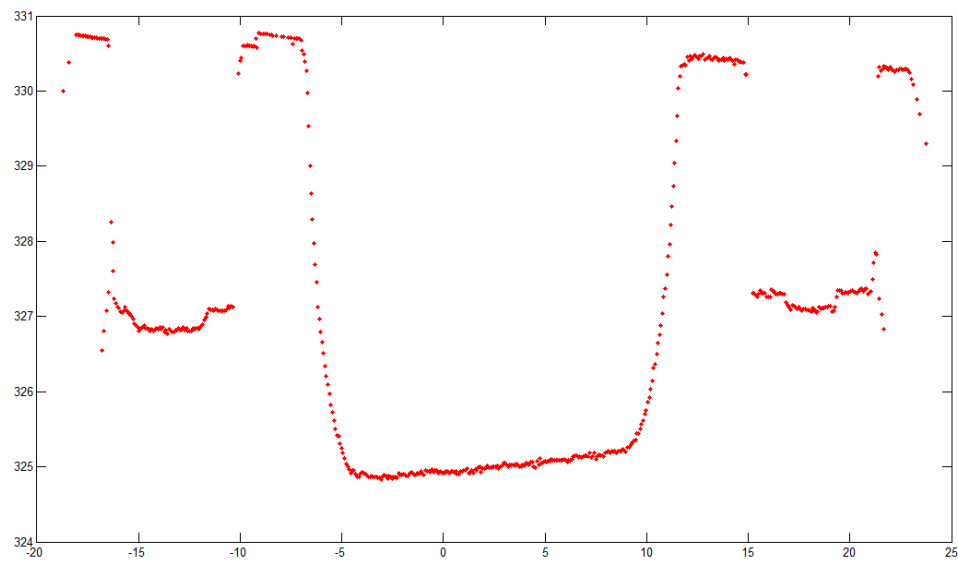


Figura 36. El mismo perfil que el de la Figura 30 recortado a los puntos mayores que 318

Código en Matlab

```
clc, clear, clf
M=xlsread('C:\Users\HP\Documents\MATLAB\Prueba1\Perfil_0001.xls');
x=input('¿A partir de que altura quieres recortar?: ');
n=size(M);
j=1;
for i=1:n(1)
    if M(i,2)>x
        Mx(j)=M(i,1);
        My(j)=M(i,2);
        j=j+1;
    end
end
plot(Mx,My, '.r'), hold on
%plot(M(:,1), M(:,2), '.b')
```

REFERENCIAS

- [1] F. Xi, C. Shu, CAD-based path planning for 3-D line laser scanning, *Computer-Aided Design* 31 (1999), pp. 473–479.
- [2] A. Bernard, M. Veron, Analysis and validation of 3D laser sensor scanning process, *Annals of the CIRP* 48 (1) (1999).
- [3] E. Zussman, H. Schuler, G. Seliger, Analysis of the geometrical feature detectability constraints for laser-scanner sensor planning, *The International Journal of Advanced Manufacturing Technology* 9 (1994) , pp. 56–64.
- [4] F. Funtowicz, E. Zussman, M. Meltser, Optimal Scanning of Freeform Surfaces Using a Laser-Stripe. In: *Israel-Korea Geometric Modeling Conference*, TelAviv, Israel, February 1998, pp. 47–50.
- [5] G. Elber, E. Zussman, Cone visibility decomposition of freeform surfaces, *Computer-Aided Design* 30 (1998), pp. 315–320.
- [6] H.-T. Yau, C.-H. Menq, Automated CMM path planning for dimensional inspection of dies and molds having complex surfaces, *International Journal of Machine Tools and Manufacturing* 35 (6) (1995) , pp. 861–876.
- [7] C.-P. Lim, C.-H. Menq, CMM feature accessibility and path generation, *International Journal of Production Research* 32 (1994), pp. 597–618.
- [8] S.N. Spitz, A.J. Spyridi, A.A.G. Requicha, Accessibility analysis for planning of planning of dimensional inspection with coordinate measuring machines, *IEEE Transactions on Robotics and Automation* 15 (4) (1999), pp. 714–727.
- [9] T. Varady, R.R. Martin, J. Cox, Reverse engineering of geometric models—an introduction, *Computer Aided Design* 29 (4) (1997), pp. 255–268.
- [10] K.H. Lee, H. Park, S. Son, A framework for laser scan planning of freeform surfaces, *International Journal of Advance Manufacturing Technology* 17 (2001) , pp.171–180.
- [11] J. M. Sánchez Muñoz, Hamilton y el Descubrimiento de los Cuaterniones, G. I. E. Pensamiento Matemático, (2011).

- [12] Micro – Epsilon. Catálogo Scan Control. www.micro-epsilon.com
- [13] Robot ABB. <http://www.abb.com/product/>
- [14] N. E. Leal, E. A. Leal, J. W. Branch, Streamlining robust clouds points using PCA and genetic algorithms. Avances en Sistemas e Informática. Vol. 6 Número 3. (2009) pp. 45 – 50.
- [15] J. Pérez et al. Mejora al algoritmo de agrupamiento k – means mediante un nuevo criterio de convergencia. 2o. Taller Latino Iberoamericano de Investigación de Operaciones. Acapulco, Gro. México (2007)